Moving task allocation and reallocation method based on body expansion behaviour for distributed multi-robot coordination

Guanghui Li*

Department of Precision Engineering,
Graduate School of Engineering,
The University of Tokyo,
Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8656, Japan
E-mail: guanghui@robot.t.u-tokyo.ac.jp
*Corresponding author

Yusuke Tamura

Faculty of Science and Engineering,
Chuo University,
Kasuga 1-3-27, Bunkyo-ku, Tokyo, 112-8551, Japan
E-mail: tamura@mech.chuo-u.ac.jp

Atsushi Yamashita and Hajime Asama

Department of Precision Engineering,
Graduate School of Engineering,
The University of Tokyo,
Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8656, Japan
E-mail: yamashita@robot.t.u-tokyo.ac.jp
E-mail: asama@robot.t.u-tokyo.ac.jp

Abstract: The inconvenience and cost of utilising existing task assignment approaches to resolve dynamical mobile task allocation. For such new domain, we first propose a method, called dynamical-sequential task allocation and reallocation, by implementing multi-round negotiation and body expansion behaviour. Every former half time step, robots negotiate sequentially and select tasks to perform, and declare the information to other robots. When all robots have finished first time selection, then the remaining unselected robots choose the remaining unassigned tasks again sequentially at the latter half time step. We set two distance thresholds for robot decision-making to apply body expansion behaviour. The advantages of our methodology are demonstrated by comparison with existing algorithms, simulation results demonstrate that the efficiency for whole system to accomplish given tasks is improved by utilising our approach. Moreover, it is more conducive to reduce the numerous computational time and communication compared with existing investigated task assignment methods.

Keywords: distributed multi-robot coordination; mobile task allocation; task reassignment; body expansion behaviour.

Reference to this paper should be made as follows: Li, G., Tamura, Y., Yamashita, A. and Asama, H. (2012) 'Moving task allocation and reallocation method based on body expansion behaviour for distributed multi-robot coordination', *Int. J. Mechatronics and Automation*, Vol. 2, No. 4, pp.240–261.

Biographical notes: Guanghui Li is currently a PhD student of the Department of Precision Engineering at the School of Engineering, The University of Tokyo. He received his Bachelor's in Traffic and Transportation from the Faculty of Transportation Engineering, Kunming University of Science and Technology, China, in 2005, and MS from the Department of Automobile Engineering at College of Mechanical Engineering, the State Key Laboratory of Mechanical Transmission, Chongqing University, China, in 2008. His current research interests include multiple mobile robot coordination, service robots, and optimisation algorithms.

Yusuke Tamura received his BE from School of Engineering, The University of Tokyo in 2003. He received his ME and PhD from the Graduate School of Engineering, The University of Tokyo, in 2005 and 2008, respectively. From 2006 to 2008, he was a JSPS Research Fellow. From 2008 to 2012, he was a Project Researcher with The University of Tokyo. He is currently an Assistant Professor of Faculty of Science and Engineering, Chuo University. His research

interests include mobile robots coexisting humans, human-robot interaction, and human motion analysis. He received the Best Paper Award at the 13th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN) in 2004, IEEE Robotics and Automation Society Japan Chapter Young Award at the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) in 2005, and Outstanding Paper Award of the Sixth International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) in 2009.

Atsushi Yamashita received his BE, ME, and PhD degrees from the Faculty of Engineering, The University of Tokyo in 1996, 1998, and 2001, respectively. From 2001 to 2008, he was an Assistant Professor of Shizuoka University. From 2006 to 2007, he was a Visiting Associate at the California Institute of Technology. From 2008 to 2011, he was an Associate Professor of Shizuoka University. From 2011, he has been an Associate Professor of The University of Tokyo. His research interests are robot vision, image processing, multiple mobile robot system, and motion planning. He is a member of ACM, IEEE, JSPE, RSJ, IEICE, JSME, IEEJ, IPSJ, ITE, and SICE.

Hajime Asama received his DrEng in Engineering from the University of Tokyo in 1989. He became a Professor of School of Engineering, the University of Tokyo in 2009. He is the Vice-president of Robotics Society of Japan since 2011. He was an AdCom member of IEEE Robotics and Automation Society from 2007 to 2009, an Editor of *International Journal of Intelligent Service Robotics*, *Journal of Field Robotics*, and *Journal of Robotics and Autonomous Systems*. He is a Fellow of JSME since 2004 and RSJ since 2008. His main research interests are distributed autonomous robotic systems, ambient intelligence, service engineering, and Mobiligence, and service robotics.

1 Introduction

The field of distributed multi-robot coordination has received increasing attention in recent decades. Many potential advantages of distributed multi-robot coordination exist in comparison with a single robot, including reduction of the complexity of the robot structure, and decreasing total system costs by implementation of multiple simple and cheap robots as opposed to a single, expensive, and complex robot. Moreover, the inherent complexity of certain tasks might require the use of multiple robots because demands of tasks are often quite difficult for a single robot to resolve. Multiple autonomous mobile robots are also assumed to enhance system robustness and flexibility by taking advantage of inherent parallelism and redundancy.

For numerous applications, distributed multi-robot coordination is useful effectively to accomplish assigned tasks by executing them concurrently. Many real world problems necessitate the use of a group of robots to accomplish a set of tasks, although difficulty arises in coordinating all of these robots to perform such a set of tasks. Previously, classifications of two kinds were proposed to solve multiple tasks assignment problem to multiple autonomous mobile robots, which named centralised task allocation and distributed task allocation. Centralised task assignment method which one robot (leader) coordinates other robots to accomplish the specified tasks optimally. The problem is optimal coordination, which is computationally difficult because the best-known algorithms present exponential progression in complexity according to their size. Another disadvantage is that the centralised task assignment method is a highly vulnerable system; if the leader agent malfunctions, then the entire system is disabled unless a new leader robot is made available.

Distributed task assignments address problems arising from centralised task allocation. Each robot coordinates with others to execute the assigned tasks. The whole system's performance no longer depends on a single leader robot. Therefore, the distributed multi-robot system becomes more robust and flexible. Additionally, robots are better able to respond to an unknown and dynamical environment because each robot can perceive its local environment independently. It is considered that distributed task allocation can reduce computational time and communication costs compared with centralised task allocation.

Nevertheless, the number of required communication costs which make use of distributed task allocation approach remains excessively high, and consumes too much computational time to obtain an optimal solution. For that reason, multi-robot coordination systems are unable to keep up with the real-time execution demands. Therefore, neither communication costs nor computational times are desirable for realistic task assignment and reassignment applications, especially for mobile tasks (for example, applications of guidance robots in exhibitions and museums, human can move randomly before robots guide them.), for which positions change randomly before the assigned robots to execute them, and the requirements of these tasks can vary over time. That is true because of the last solution, by which robots have been assigned to given tasks, might not be suitable for current circumstance when conditions are changing over time. The system should reallocate robots to tasks to find the potentially optimal solutions. For such a new domain, we propose a dynamical-sequential moving task allocation and reallocation method for distributed multi-robot coordination system based on multi-round negotiation and body expansion behaviour. The word 'dynamical' means that both the number of tasks and the requirements of tasks can change, whereas 'sequential' indicates that one robot assigns tasks after another robot under some order. As described in this paper, we use the proposed approach mainly to improve the accomplished efficiency for the whole distributed multi-robot coordination. Moreover, it is more conducive to reducing the numerous computational times and communication costs compared to existing investigated task assignment methods.

The remainder of this paper is structured as follows. The next section presents discussion of the related works of the well-known field task allocation. Section 3 presents a formal definition of moving task assignment problem, and presents discussion of the disadvantages of existing methods in addressing our defined problem. The notion of body expansion behaviour is described in Section 4, which sets two thresholds for robots to make decisions. We also detail the proposed task allocation and reallocation algorithm in this section. Section 5 presents discussion of simulation results, which compares our approach to the existing general task allocation approach. Finally, Section 6 presents a description of conclusions and sketches a prospective plan of future work.

2 Related works

The task allocation problem for a multi-robot coordination system is a widely studied field. It is classifiable broadly into two classes: one is centralised planner-based systems, for which planners are often based on auction mechanisms in which robots bid for tasks, e.g., Gerkey and Mataric's (2002) MURDOCH. Wawerla and Vaughan (2010) proposed a method for team-task allocation in a multiple robots transportation system because systems of such kinds have agents and tasks that is still fixed. Moreover, capabilities and resources are independent of time, although in real world applications, it is not useful. Another problem is a system which relies on individual robots to make individual task allocation decisions without considering other team members and optimisation of the whole system. Empirical results of an auction-based algorithm for dynamic allocation of tasks to robots were proposed by Nanjanath and Gini (2010). From their research, they proposed a method of repeated auctions for distributing tasks dynamically among a group of cooperative robots. The distinctive feature of this algorithm is its robustness to uncertainty and to robot malfunctions that occur during task execution.

A method of another kind is distributed task assignment, such as the methods described by Asama et al. (1992) and Ozaki et al. (1997), who develop an autonomous and decentralised robot system called ACTRESS to address issues of communication, task assignment, and path planning among heterogeneous robotic agents. This approach revolves primarily around a negotiation framework that allows robots to recruit help when needed. Parker (1992, 1994a, 1994b, 1994c, 1997, 1998) formulated a related multi-robot task allocation problem called the ALLIANCE efficiency problem. Werger and Mataric (2000) introduced a broadcast of local eligibility (BLE) approach to multiple robots coordination. The BLE mechanism involves a comparison of locally determined eligibility with the best eligibility calculated using peer behaviour on another robot. A distributed multi-robot cooperation framework for real-time task achievement was proposed by Sariel and Balch (2008a), and Sariel et al. (2008b). The framework integrates a distributed task allocation scheme, coordination mechanisms and precaution routines for multi-robot team execution. When initial assignments of tasks might become inefficient during real-time execution because of real world issues such as failures; these allocations are subject to change if efficiency is an important concern. Reallocations are needed and should be performed in a distributed fashion. They proposed an online dynamic task allocation system for reallocation to achieve a team goal that can respond to and recover from real-time contingencies. Parker and Tang (2006), and Tang and Parker (2007) presented a reasonable system that enables a group of heterogeneous robots to form coalitions to accomplish a multi-robot task using tightly coupled sensor sharing. The advantages of this new approach are that it enables robots to synthesise new task solutions using fundamentally different combinations of sensors and effectors for different coalition compositions. Moreover, it provides a general mechanism for sharing sensory information across network robots. However, all the points presented above mainly relate to the computational performance. Tasks are static: they do not describe dynamical tasks and methods of task reassignment. Furthermore, they do not discuss fault tolerance, flexibility, and robustness. Moreover, when a robot fails, the system does not know how to address it.

Other related works examined task allocation problems such as a coalition maintenance scheme for dynamic reconfiguration of assigned tasks to obtain optimum allocations in noisy environments during the running time (Sariel and Balch, 2005a, 2005b). This framework is used to address different types of failures that are common in robot systems and to solve conflicts in cases of communication and robot failures. Task allocation using particle swarm optimisation method is suggested to determine coalitions and sequences for all targets (Sujit et al., 2008). Sujit et al. employ this algorithm to resolve the problem in a reasonable amount of time. Market-based auction (Dias et al., 2006; Stentz and Dias, 2006) is well known for dealing with task allocation problems: the system auctions tasks to all robots. After bidding for tasks, robots that obtain profits that are largest for the whole system execute these tasks. Additionally, they investigate a real-time single-item allocation method under an uncertain and dynamic environment (Sariel and Balch, 2005a, 2005b). The initial assigned targets might have to be reallocated during a time when the environment is dynamic and/or unknown. The market-based auction method can be successful and effective to resolve the conventional task allocation domain, large number of researchers improve and study the variation of such method, such as sequential single-item auction (Zheng et al., 2006), distributed sequential auction (Sujit and Beard, 2007), and decentralised task sequencing method (Paola et al., 2011).

3 Task description

3.1 Formal definition

This paper describes task assignment problems among multiple, fully distributed, initially homogeneous mobile robots. We develop a novel method of task allocation and reallocation that can deal with dynamical moving tasks. The formal definition of this problem is that we assume environment of such kind included missions of two kinds: one is the initial mission, the initial mission is that assign multiple mobile tasks to robots reasonably and efficiently; another is the final mission, with such a system that a robot guides a task from the initial position to the destination which tasks should reach.

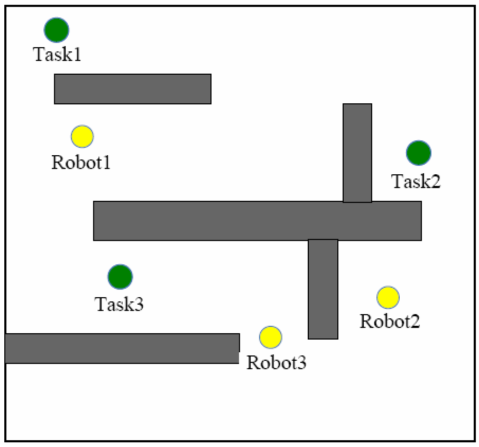
For the initial mission, because tasks move randomly before they are assigned to robots to execute them, and because the conditions of these tasks can vary over time, we should assign and reassign tasks to robots properly: we allow sets of tasks T and robots R to be time-dependent at every instant of time or over the entire history (i.e., $T(t)$, $R(t)$) and require that the objective functions be minimised/maximised (the task allocation method should minimise objective functions, which are cost, energy, and others. Reversely, it should maximise the objective functions which are efficient and so on.), the definition also covers online and dynamical domain where tasks and robots might be added or removed over time. We propose a dynamical-sequential task allocation and reallocation methodology based on body expansion behaviour to resolve domains of this kind. As the final mission, when robots move to nearby tasks, tasks transmit its destinations to robots, then in each robot global coordinate system, robots find the positions of robots' destination, plan the optimal path and guide tasks to the destinations.

3.2 Disadvantages of existing methods

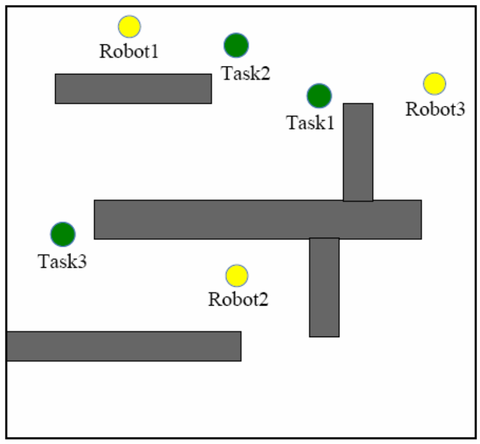
Few researchers have addressed the domain of tasks which are dynamical and which are arbitrarily movable. All existing methods are suitable for tasks in which positions are fixed. When we use existing methods to solve mobile task assignments, the whole system will become extremely inefficient. Furthermore, earlier reports neglect discussion of task reallocation when robots are executing tasks, except for robot malfunction, partial system failure, and communication failure. Actually, for mobile tasks in terms of position and requirement change, we should not only find an available task assignment solution. We should also develop a mode by which robots perform tasks efficiently for the overall coordinated system.

For example, as described in this paper, if we consider homogeneous robots, then the efficiency for robots to perform tasks depends on the time needed by which robots reach the task location. This measure presents the task and robot's position, which is a function of time. Therefore, the efficiency varies with time. Robots should therefore select the optimal tasks for which the necessary time is the shortest (i.e., the distance between robots and tasks are shortest.) to perform every time to improve the efficiency. As Figure 1 shows, at $Time_0$, the system assigns task1 to $robot_1$, $task_2$ to $robot_2$ and $task_3$ to $robot_3$ according to the shortest distances for robot positions and tasks. At $Time_1$, since changing positions of tasks the system should reallocate tasks to robots reasonably, the values of distance between $robot_1$ and $task_1$, $robot_2$ and $task_2$, $robot_3$, and $task_3$ are greater than the distance between robot1 and $task_2$, $robot_2$ and $task_3$, and $robot_3$ and $task_1$.

Figure 1 Dynamical moving task allocation and reallocation, (a) at $Time_0$ (b) at $Time_1$ (see online version for colours)



(a)



(b)

Notes: (a) At $Time_0$, tasks T_1, T_2, and T_3 are assigned respectively to robots R_1, R_2, and R_3. (b) At $Time_1$, because of the changing positions of robots and tasks, T_1, T_2, and T_3 are reassigned respectively to robots R_3, R_1, and R_2.

Few previously reported approaches explicitly address the problem of minimising communication costs, computational times, and memory. For example, all market-based auction methods, ALLIANCE, and BLE need each robot plans path from location of itself to each task, calculates distances between robots and tasks, when the positions of tasks change. Once the situations of tasks and robots vary, systems should auction these tasks for all robots. After bidding tasks, robots which obtained profits are largest for the whole system execute these tasks. In other words, the efficiency of these methods is extremely low to address dynamical moving task allocation and reallocation problems. It takes a long computational time to motion planning, distance calculation and tasks negotiation. Both BLE and ALLIANCE methods do not consider global efficiency explicitly, although these methods are satisfied with finding any feasible solution. A notable exception is the work by Nanjanath and Gini (2010), where they propose a method of repeated auction for distributed tasks dynamically among a group of cooperative robots. Tasks that are not yet achieved are re-submitted for bids every time a task has been completed. The repeated auction comes closest to our approach. Main differences include our proposed system reallocation tasks for robots at every time step. Then we use body expansion behaviour to reduce communication costs and computational times for each robot when the distance between robot and task is less than a given threshold.

In several reports, Turra et al. (2004a, 2004b) first introduce an algorithm for allocation at mission-time of moving targets to a group of unmanned vehicles (UAV). The Hungarian algorithm is implemented to perform optimal task assignment; then exact path lengths between vehicles and targets are computed through the off-line computed Dijkstra paths. For dynamical mobile task allocation and reallocation method of distributed multi-robot coordination, we propose dynamical-sequential task allocation and reallocation. This approach implements multi-round negotiation and body expansion behaviour for robots to select tasks. To implement body expansion behaviour, we set two distance thresholds for robot decision-making. Based on body expansion behaviour, one robot can request, accept, and refuse other robots' requests to execute tasks by intention communication. Herein, we demonstrated that this method is an approximate global optimal assignment method and that it expends acceptable communication costs and computational times compared to existing investigated task assignment methods.

4 Task allocation and reallocation method

4.1 Mathematical model

As described above, we only consider a homogeneous set of robots. The efficiency for distributed multiple robots coordination system consists of two important evaluations. One is the *summation executed costs of all robots* – E_{SEC} by which robots perform all mobile tasks. E_{SEC} depends on the relative positions of tasks and robots. In other words, it depends on the *summation completion time of all robots* – T_{SCT} necessary for robots to reach the task location, it is a function of time. Because all tasks can move randomly before they are assigned robots to execute, E_{SEC} and T_{SCT} for which a robot performs a task varies. For that reason, robots must select optimal tasks for which the needed executed costs by robot are least to perform. Doing so for each task improves the overall system efficiency.

Another important evaluation is the time at which the last task is completed, which we define as *last task completion time* – T_{LTC}. We know that we can declare that the entire system is completed only after the last task is finished. In some situations, the system consumes very little E_{SEC} and T_{SCT}, whereas T_{LTC} might be large compared with other situations. The salient meaning is that robots take a long time to execute the last task in these situations. Therefore, we say that the time at which entire system is completed is later than others, although the E_{SEC} and T_{SCT} are more efficient. Actually, such a situation arises frequently in coordination system which uses dynamical-sequential task allocation and reallocation method.

The locations of M robots $V_R \in \{R_1, R_2, \cdots, R_m\}$ is the number of robots) and N mobile tasks $V_T \in (T_1, T_2, \cdots, T_n)$ (n is the number of tasks) are known, as is the cost function $E_{SIR,Ri}$ (where $i \in \{\text{M}/1, 2, \cdots, m\}$) that specifies the i^{th} *summation individual robot cost* when the whole system is completed. $E_{Cost,Ri,t\text{–}time}$ specifies the *cost of i^{th} robot* from t time step to $t + 1$ time step. The objective is to find an allocation of tasks to robots such that the total cost E_{SEC} is minimised for the whole system. Because we only consider a homogeneous set of robots and tasks, the major criterion for the proposed strategy is to optimise the total *travelled distance of all robots* – D_{TTD}. Accordingly, we can use the i^{th} *summation individual robot distance* – $D_{SIRD,Ri}$ and the *distance of i^{th} robot* – $D_{Distance,Ri,t\text{–}time}$ from t time step to $t + 1$ time step, to denote $E_{SIR,Ri}$ and $E_{Cost,Ri,t\text{–}time}$, respectively. The model formulated to enhance the mobile task allocation and reallocation is presented below. Let V_R denote the set of robots and V_T denote the set of mobile tasks.

The objective functions are to minimise

$$E_{SEC} = \sum_{Ri \in V_R} E_{SIR,Ri} = \sum_{Ri \in V_R} D_{SIRD,Ri} \tag{1}$$

where

$$D_{SIRD,Ri} = \sum_{Ri \in V_R} \sum_{st} D_{Distance,Ri,t-time} x_{Tj}^{Ri}$$

$$\forall st \in (0, +\infty), \quad j \in \text{N} \tag{2}$$

$$T_{SCT} = \sum_{Ri \in V_R} \sum_{st} T_{TimeStep,Ri} x_{Tj}^{Ri} \tag{3}$$

$$T_{LTC} = \max\left\{T_{Time,Tj}, \quad Tj \in V_T\right\} \tag{4}$$

subject to the following equations.

$$x_{Tj}^{Ri} = \begin{pmatrix} 1 & Ri \text{ selects } Tj \\ 0 & \text{Others} \end{pmatrix} \tag{5}$$

$$\sum_{Tj \in V_T, Ri \in V_R} x_{Tj}^{Ri} \le M \tag{6}$$

$$x_{Tj}^{Ri} \in \{0, 1\} \quad \forall Ri \in V_R, Tj \in V_T \tag{7}$$

$$\pi_{t-time,Tj}^{Ri} x_{Tj}^{Ri} = \begin{cases} 1 & Ri \text{ selects } Tj \\ 0 & \text{Others} \end{cases} \tag{8}$$

$$\sum_{Ri \in V_R} \sum_{Tj \in V_T} \pi_{st,Tj}^{Ri} x_{Tj}^{Ri} \le N \tag{9}$$

Therein, *one time step* – $T_{TimeStep}$ specifies a unit length of time, $T_{Time,Tj}$ is the time when the j^{th} task T_j is completed. A binary variable x_{Tj}^{Ri} denotes whether the i^{th} robot $Ri \in V_R$ performs the j^{th} task Tj selected from all tasks V_T. $T_{TimeStep,Ri}$, signifies the number of time steps that robot $Ri \in V_R$ selects task $Tj \in V_T$; $\pi_{t-time,Tj}^{Ri}$ and $\pi_{st,Tj}^{Ri}$ are a binary value showing whether task $Tj \in V_T$ is executed at time step *t-time* and all time steps, respectively.

The objective function, equation (1) minimises the execution cost of the whole distributed multi-robot coordination system.

In this case, the system cost is the total travelled distance that the robots move. Equation (4) minimises the completed time of the last task, which is to minimise the time to finish the whole system. The first set of constraints, equation (5), specifies that each robot performs exactly one task. The second set of constraints, equation (8), specifies that each task is assigned to exactly one robot at each time step.

4.2 Body expansion behaviour

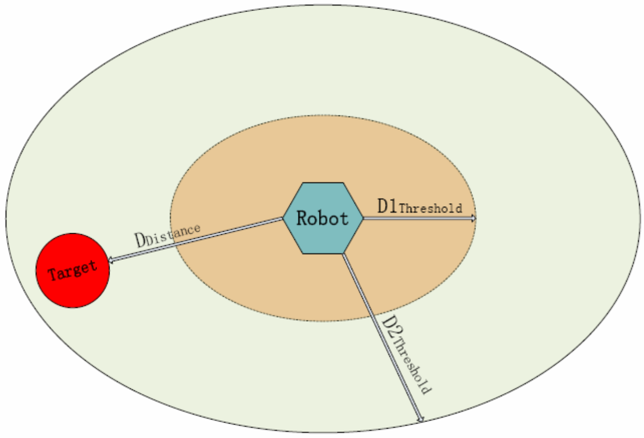
The notation body expansion behaviour means that a robot can transmit its own intention and the receiver executes its requirement. Thereby, a robot can control others' behaviour by intention transmission using communication (Fujiki et al., 2007). This demonstrates an expansion of the robot's degrees of freedom (DOF). A multi-robot coordination system can improve flexibility and adaptability by application of such body expansion behaviour.

Two distance thresholds for robot decision-making are set to implement body expansion behaviour. One is the small distance threshold $D1_{Threshold}$, which means that the robot is about to execute the assigned task. Another is the large distance threshold $D2_{Threshold}$, which means that robots have a long time to execute the assigned task (Figure 2). If the distance is greater than the $D2_{Threshold}$, then a robot can request that other robots execute the assigned task. If the distance between $D1_{Threshold}$ and $D2_{Threshold}$, then robots compare the distances and select the tasks presenting the shortest distance. If the distance is less than $D1_{Threshold}$, then robots refuse all others' requests. All robots exist in one of three working states:

1. *free-robot*, when the robot has not been assigned a task
2. *half-free-robot*, when the robot has been assigned a task but is not executing the task, or the distance is less than $D2_{Threshold}$, but greater than $D1_{Threshold}$
3. *busy-robot*, when a robot is executing a task, or the distance is less than $D1_{Threshold}$.

When robots find remaining unguided tasks and free-robots exist in the environment, then the robot can request that the free-robot guide the remaining unguided tasks.

Figure 2 Distance threshold (see online version for colours)



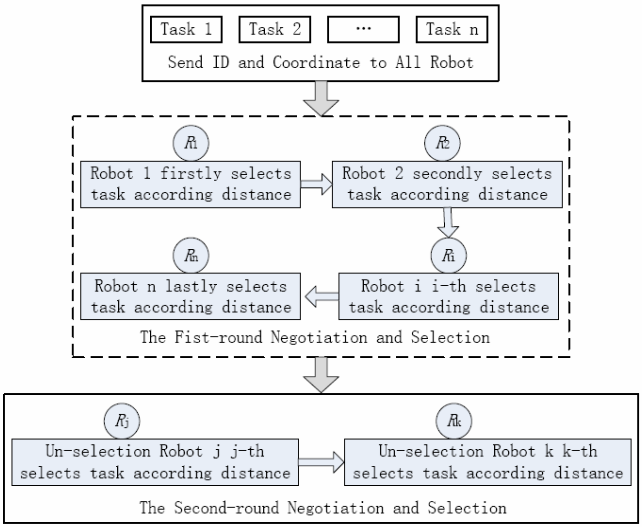
Notes: Two distance thresholds are set for robot decision-making. If distance $D_{Distance}$ is greater than $D2_{Threshold}$, then the robot requests that other robots execute the task. If $D_{Distance}$ between $D1_{Threshold}$ and $D2_{Threshold}$, then the robot compares the distance and selects the shorter distance task to execute. If the $D_{Distance}$ is less than the $D2_{Threshold}$, then robot refuses the others' request.

4.3 Proposed task allocation and reallocation algorithm

Assuming that all robots are homogeneous robots with the same speed, function, and structure, and that they can mutually communicate using radio frequency broadcast, then one robot allocates only a single task at a time, executes only a single task, and guides the assigned task to its destination.

The tasks are distributed randomly in the environment, and can move anywhere with varied speed before robots reach around them. Each task does not know the location of destination unless under the robot guided to it. Furthermore, all tasks await guidance in the priority queue under the principle of 'first in – first executed'. A robot always executes the relative highest priority task irrespective of the other tasks move around it. We propose a novel task allocation method that can reallocate tasks to robots according to the shortest distance. In the environment, $Ri \in \{R_1, R_2, \cdots, R_m\}$ denotes the i^{th} robot, $Tj \in \{T_1, T_2, \cdots, T_n\}$ denotes the j^{th} task, The D_{RiTj} denotes the utilisable distance from Ri to Tj, and $n \ge m$. In the initial state, the working statuses of all robots are free-robot, and wait for executing tasks (Li et al., 2011a).

Figure 3 Illustration of the initial step (see online version for colours)



Tasks broadcast request information including task IDs and coordinates to all robots at every time step. In the initial time step (Figure 3), there are two rounds of negotiation and selection for each robot. For the first round, all robots receive request information from tasks, then plan paths to all tasks and calculate the distances in the robot's global map. Robots are assigned priority according to the robot ID, the priority of robot which the ID is small is larger than the priority of a robot which the ID is large. Robots $R1$ – robot Rm select tasks to perform according to the given distance thresholds sequentially. If there are distances which between robots and tasks are less than $D1_{Threshold}$, robots select the task to perform which present the smallest distance. Otherwise, robots select no task, and others are requested to execute it. Then all robots declare the selection information to other robots. When all robots have finished the first selection, the remaining un-selection robots choose the remaining unassigned tasks again sequentially the second time round. That is based on the priority of robots' ID, the later robot's ID should receive the entire task selection information from the former robot. Then it can carry out the task-selection process, the remaining unselected robot sequentially selects the unassigned task for which the distance is shortest in the unassigned tasks to perform, even though the distance between them is more $D2_{Threshold}$. The algorithm of the initial time step is the following:

1 *Tasks broadcast request information including task IDs Tj and coordinate to all robots Ri.*

2 *FOR Ri (i = 1, i <= m, i++)*

3 *Ri plans a path for the first m tasks, calculates distances* $D_{RiTj} \in \{D_{RiT1}, D_{RiT2}, \ldots, D_{RiTm}, i, j, m \in M\}$ *between Ri and Tj.*

4 *Task selection model in Ri: Compare the distances* $D_{RiTj} \in \{D_{RiT1}, D_{RiT2}, \ldots, D_{RiTm}, i, j, m \in M\}$.

5 *IF Several distances* D_{RiTj} *are less than* $D1_{Threshold}$.

6 *THEN Assign Tj to Ri of which* D_{RiTj} *is shortest.*

Broadcast the selection information to other robots.

The working status of robot Ri changes to busy-robot.

7 *ELSE All of* D_{RiTj} *are more than* $D1_{Threshold}$.

Request other robots to execute the first m tasks.

8 *FOR Ri (i = 1, i <= m, i++) except busy-robot*

Task selection model in robot Ri: Compare distances $D_{RiTj} \in \{D_{RiT1}, D_{RiT2}, \ldots, D_{RiTm}, i, j, m \in M\}$.

9 *Select Tj to Ri which* D_{RiTj} *is shortest.*

Broadcast the selection information to other robots.

The working status of Ri changes to half-free-robot.

During execution by which the system has assigned all tasks to robots and time up until the next time step, the intermediate algorithm is the following:

1 *Ri plans optimal path to Ti according to global map of environment.*

2 *Ri moves along the optimal path toward Tj.*

3 *IF Ri reach the location of Tj*

4 *Ri guides Tj to its destination.*

5 *Ri sends the guiding information to other robots.*

6 *The working status of Ri changes to busy-robot.*

7 *IF Ri guides Tj to its destination.*

8 *Ri sends the report of guidance completion to all other robots.*

9 *Ri clears up the information about Tj.*

10 *Ri changes to free-robot.*

11 *IF A new requested task Tu exists,*

12 *THEN Ri plans path to Tu and calculates DRjTu.*

13 *IF DRiTu is less than* $D1_{Threshold}$,

14 *THEN Ri broadcasts the selection information to other robots.*

15 *The working status of Ri changes to busy-robot.*

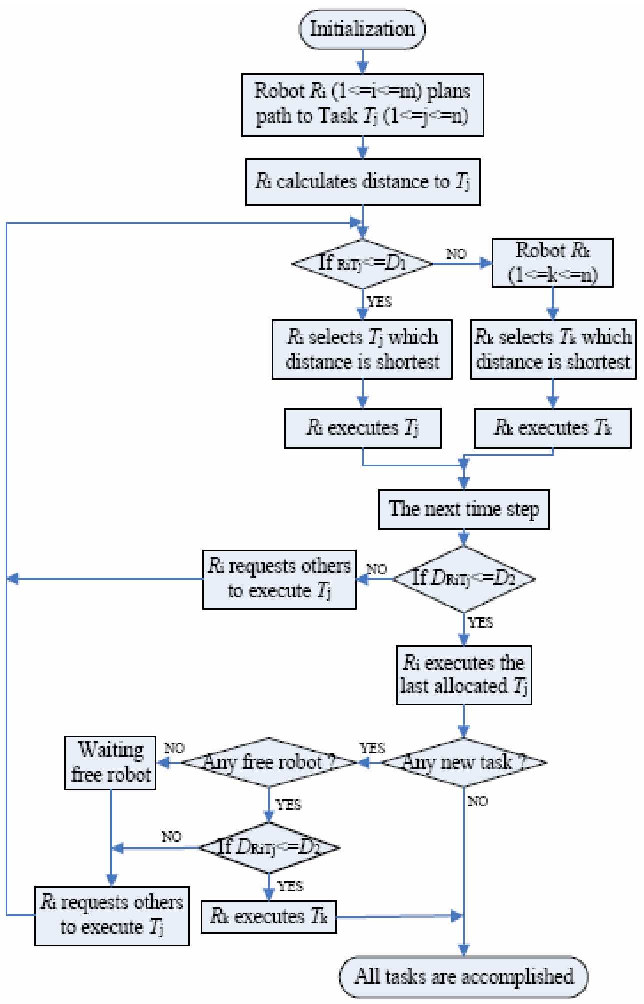
16 *ELSE The working status of robot Ri changes to half-free-robot.*

Because of the dynamical tasks can move randomly before the assigned robots to reach around and execute them, the condition of these tasks can vary over time, distances between robots and the corresponding assigned tasks might vary. Consequently, systems should reallocate tasks to robots at every time step based on using body expansion behaviour during the implemental period, to improve the efficiency of which robots execute tasks. If the distance between the robot and corresponding assigned task is greater than $D2_{Threshold}$, then the robot will request that others to execute this task and broadcasts the information. For other robots of $R1$ to Rm sequentially, it compares the distance with $D2_{Threshold}$ and selects the task for which the distance is shorter between the requested task and the latest assigned task because other robots can accept or refuse the request according body expansion behaviour. If all other robots refuse the task, then the robot should continue to select the task to perform despite the distance is greater $D2_{Threshold}$. Robots also request that other robots execute the assigned task when a robot shows failure. The algorithm of the next iterative time step is the following:

```
Ri only deals with Tj which is assigned in the prior time step.
FOR    Ri (i = 1, i <= m, i++)
           Ri plan path and calculates D_RiTj to Tj.
IF     D_RiTj is less than D2_Threshold,
       IF     D_RiTj is less than D1_Threshold,
       THEN        Ri continues to move toward Tj.
                   Ri broadcasts the selection information to other robots.
                   The working status of robot Rj change to busy-robot.
ELSE
              IF     There is a request to execute Tp from Rk,
              THEN       Compares D_RiTj and D_RiTp.
                         Selects Tp which distance D_RiTp is Shorter.
                  IF       D_RiTp is less than D2_Threshold,
                       IF      D_RiTp is less than D1_Threshold,
                               Request other robot to execute Tj.
                               Broadcast the selection information to others.
                               The working status of Ri changes to busy-robot.
                       ELSE        Request other robot to execute Tj.
                                   Broadcast the selection information to others.
                                   The working status of Ri changes to half-free-robot.
                  ELSE         Ri continues select task Tj.
                               Broadcast the selection information to other robots.
                               The working status of Ri change to half-free-robot.
    IF     The distance D_RiTj is greater than D2_Threshold,
               Ri Requests other robot to execute task Tj.
IF     All other robots refuse to execute Tj,
THEN       Robot Ri Continues select Tj.
           Ri broadcasts the selection information to other robots.
           The working status of robot Ri change to half-free-robot.
Return to the intermediate algorithm.
Until all tasks are executed or time-out.
```

The overall algorithm of our proposed novel dynamical–sequential task allocation and reallocation method is portrayed in Figure 4.

Figure 4 Our algorithm (see online version for colours)

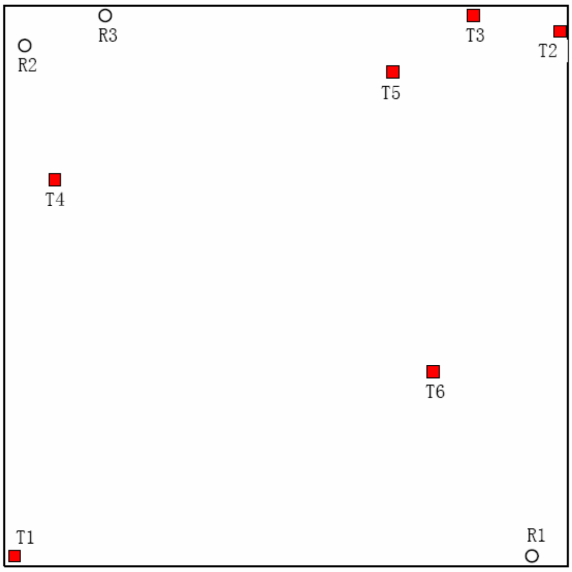


5 Simulation and results

5.1 Simulation environment setting

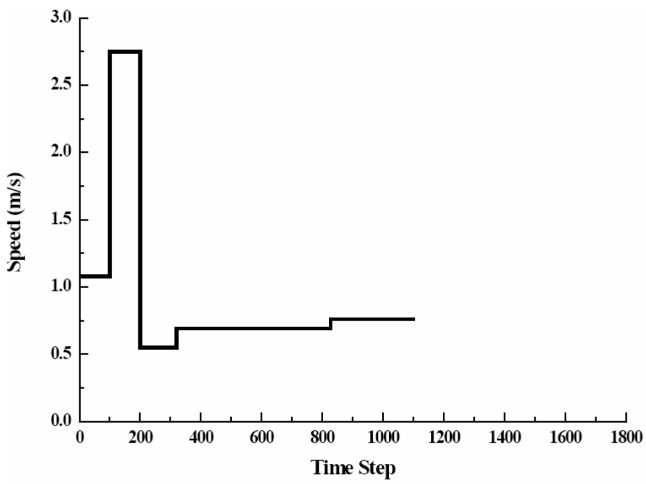
To demonstrate the validity and efficiency of our approach, various experiments were conducted using computational simulations. The simulation environment without obstacles is built up with the setting of 400 × 400 m^2. Three robots (represented by small black circles) and six tasks (represented by small red rectangles) are employed for simulation in Figure 5. At the initial time step, the first three tasks and three robots are distributed randomly in this environment. Then at time step T = 500, the fourth task moves in the environment. Similarly, at time steps T = 800 and T = 850, the fifth and sixth robots move in it. During simulation, tasks move with variable speed over time as depicted in Figure 6, whereas the speed of all robots is constant as 0.76 m/s. $D1_{Threshold}$ is 4 m; $D2_{Threshold}$ is 40 m. To compare our approach, we simulate two kinds of method: a general distributed repeated auction method and a centralised global optimal task assignment in the same situation.

Figure 5 Simulation environment (see online version for colours)



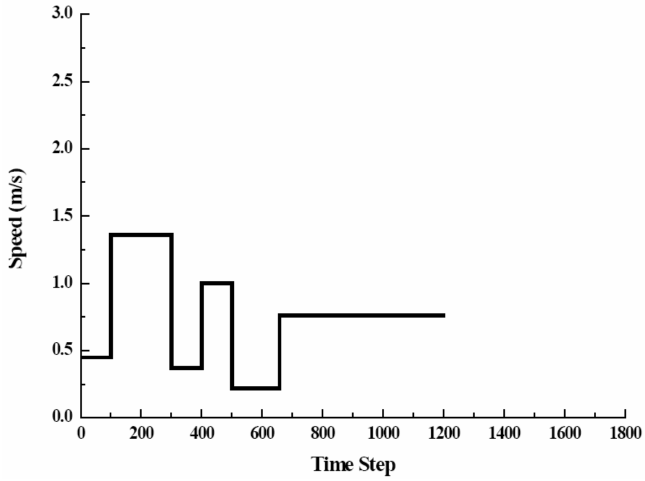
Notes: The simulation environment is built up with the setting of 400 × 400 m^2. Three robots (shown as small black circles) and six tasks (shown as small red rectangles) are used.

Figure 6 Speed of tasks, (a) speed of task *T1* (b) speed of task *T2* (c) speed of task *T3* (d) speed of task *T4* (e) speed of task *T5* (f) speed of task *T6*

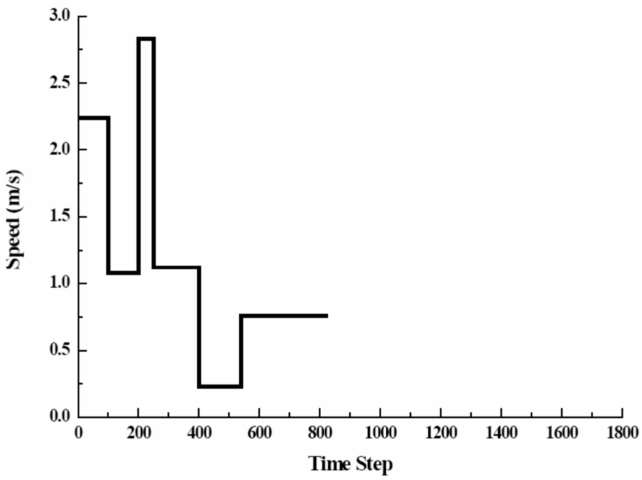


(a)

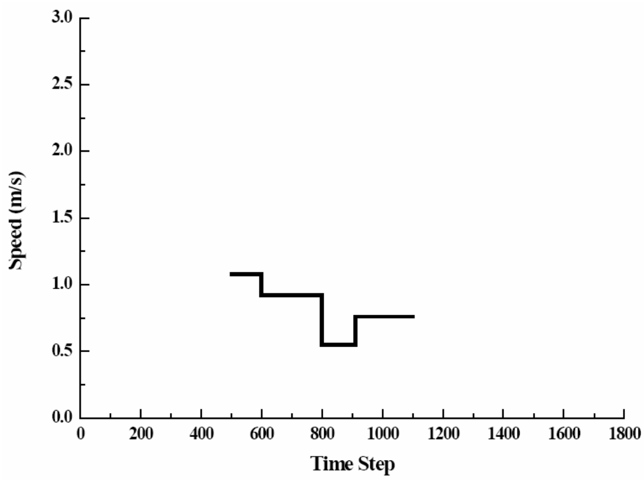
Figure 6 Speed of tasks, (a) speed of task *T1* (b) speed of task *T2* (c) speed of task *T3* (d) speed of task *T4* (e) speed of task *T5* (f) speed of task *T6* (continued)



(b)

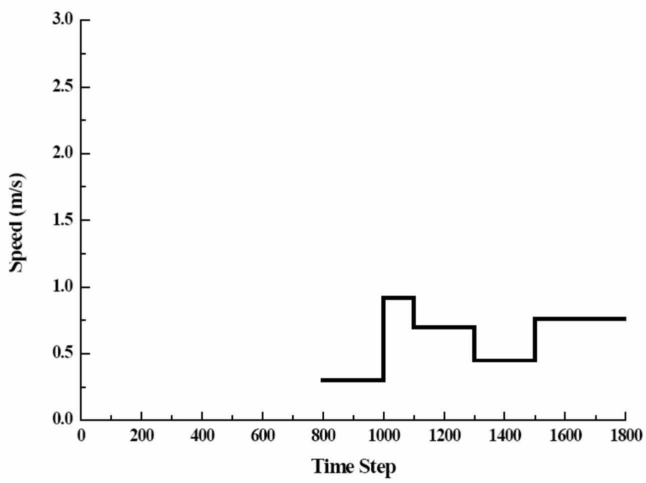


(c)



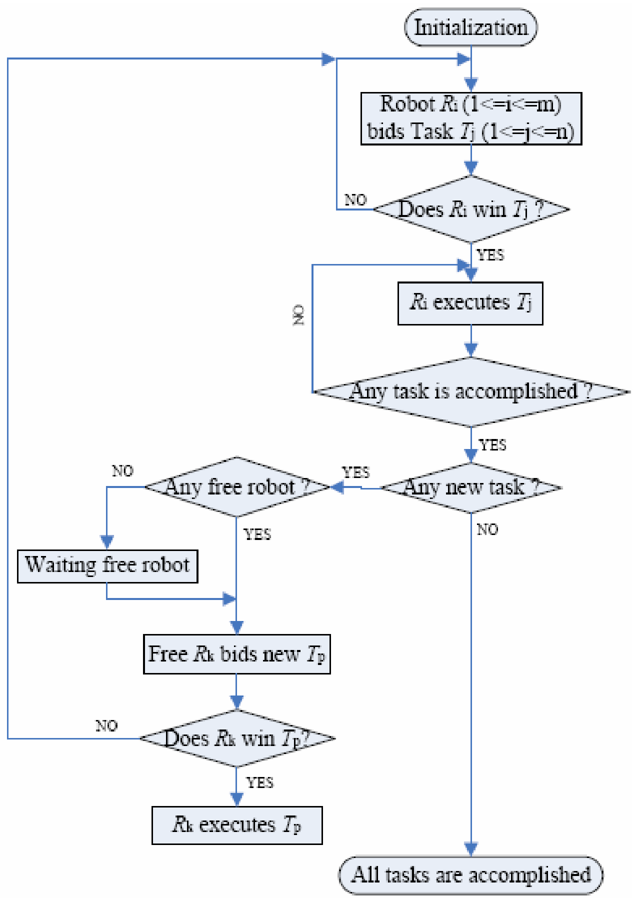
(d)

Figure 6 Speed of tasks, (a) speed of task *T1* (b) speed of task *T2* (c) speed of task *T3* (d) speed of task *T4* (e) speed of task *T5* (f) speed of task *T6* (continued)



(e)

Figure 7 The repeated auction method (see online version for colours)



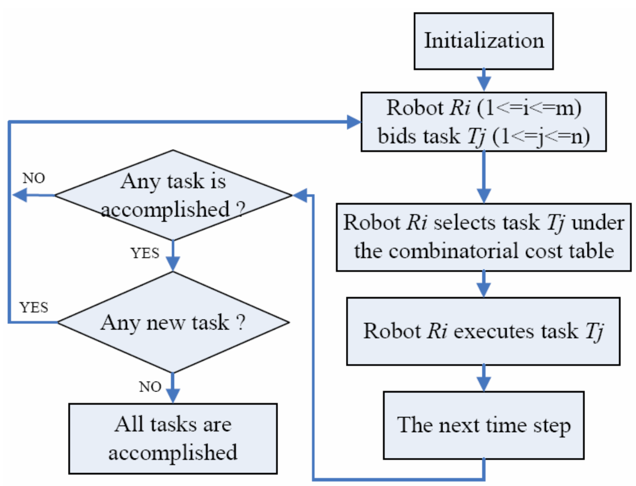
5.2 Repeated auction method

Empirical results of market-based algorithm for robots that dynamically allocates tasks to robots is proposed by Nanjanath and Gini (2010). As described in this paper, they propose a method of repeated auction for distributed tasks dynamically among a group of cooperative robots. First, robots execute tasks which are assigned initially. When each task is completed, all remaining tasks are auctioned again and reassigned to robots. Results show that the distinctive feature of their algorithm is its robustness to uncertainty and to robot malfunctions that happen during task execution when unexpected obstacles, loss of communication, and other delays might prevent a robot from completing its allocated tasks. The algorithm of the repeated auction method is portrayed in Figure 7.

5.3 Global optimal task allocation method

Global optimal task allocation method is extended from combinatorial optimisation and market-based task allocation method. It is proved that combinatorial optimisation can obtain the global optimal assignment (Li et al., 2011b). In addition, market-based task allocation is a simple and valid method for complicated assignment. Robots bid tasks and communicate costs with other robots. For each robot, makes a combinatorial cost table after congregating all the bidding from others, then selects task to execute based on objective function at every time step. The objective function which is to be minimised executed costs and maximised accomplished efficiency for the whole system. The objective of this method is to reduce the total tasks executed time for the entire system. The algorithm of global optimal allocation and reallocation approach is shown in Figure 8.

Figure 8 The global optimal method (see online version for colours)

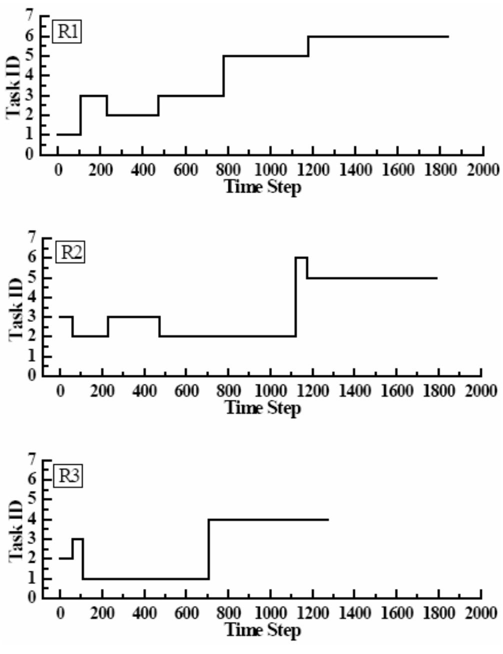


5.4 Simulation results

Three tasks enter into the environment at different positions in the initial time step. The task purposes are to reach their destinations, although not all of them know where the destinations are. Therefore, tasks request robots to guide them to their destinations. However, during the time that robots reach around (approach) tasks, the tasks can move randomly instead of standing in the specified location when waiting.

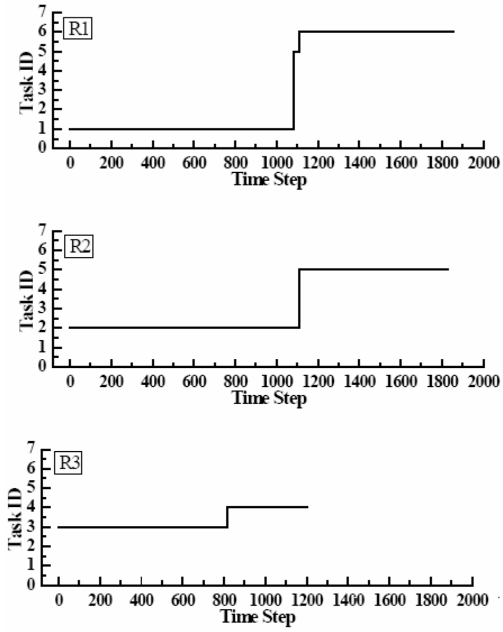
Figure 9 portrays the selected situations of robots that use the approaches described above at every time step. Each robot compare the large/small distance threshold with distance $D_{Distance}$ which is from the location of itself to the task. If $D_{Distance}$ is greater than $D2_{Threshold}$, then the robot requests other robots to execute the task; otherwise, the robot executes the task by itself. We employ Figure 8(a), which is related to our approach, as an example. At time steps $T = 62$, 110, 230, 475, and 1,179, tasks are reallocated to robots because the distances between them are greater than $D2_{Threshold}$. At $T = 371$, robot *R*3 arrives at *T1* and will guide *T1* to destination *D1*. In such a situation, *R3* will refuse all requests from other robots because the distance is less than $D1_{Threshold}$, $T = 507$, 667, 969, 1,422, and 1,448 are the same as $T = 371$. *T4* walks into the environment at $T = 500$ (the same as $T = 800$ and 850 are distributed into the environment). *T4* will move randomly under the unassigned state because each robot can be assigned to only a single task to guide each time until there is a free-robot that like $T = 705$, *T1* has arrived at *D1* under the *R3* guiding, in the next time robot will check whether there is an unassigned task. The robot will be assigned to the unassigned task if an unassigned task exists in the environment such as $T = 705$, 1,120, 1,273 and 1,789, or as in a situation where that robot will move freely because there is no unassigned task (as $T = 782$). The snapshots are presented in Figure 10.

Figure 9 Selected situations, (a) proposed method (b) repeated auction method (c) global optimal method

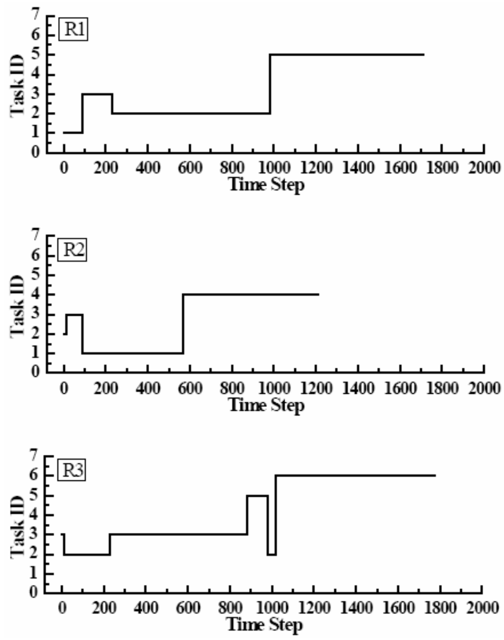


(a)

Figure 9 Selected situations, (a) proposed method (b) repeated auction method (c) global optimal method (contniued)



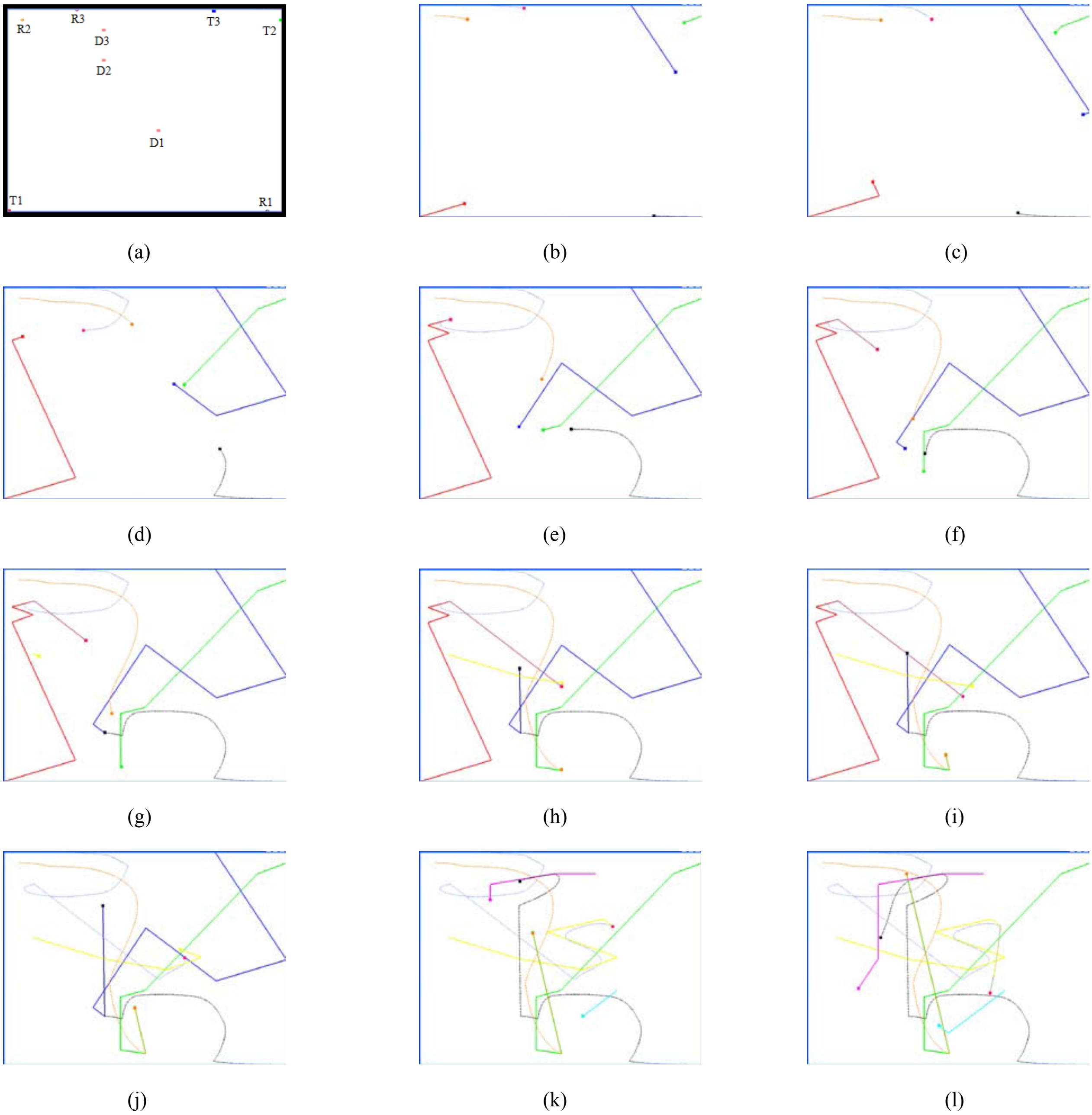
(b)



(c)

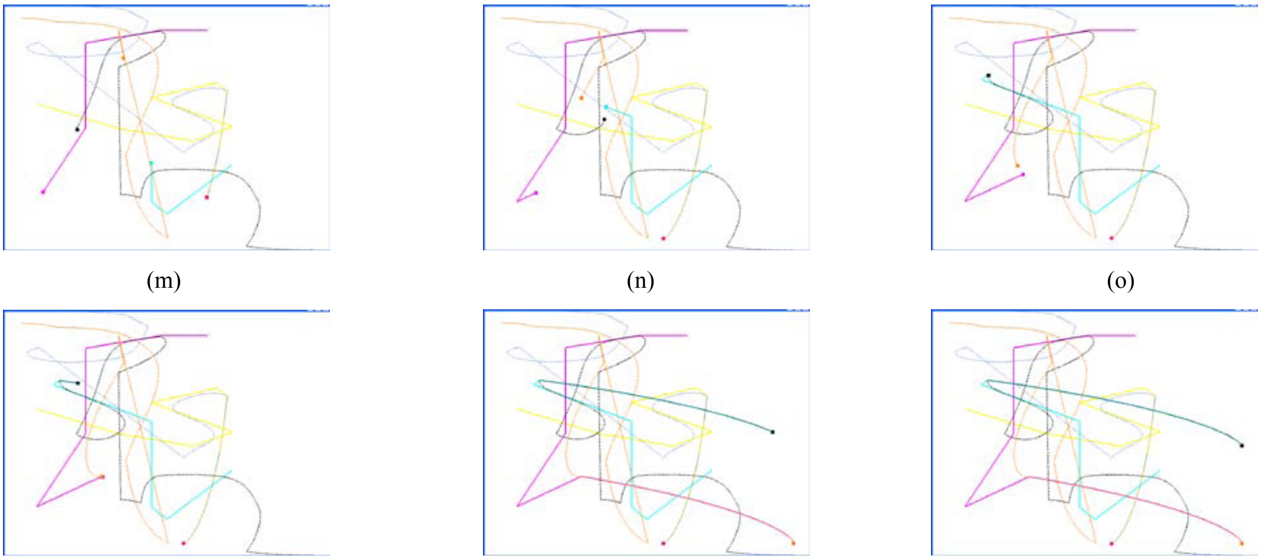
Nevertheless, under the same simulation condition [Figure 8(b)], at the initial time step, tasks *T1*, *T2*, and *T3* are allocated respectively to robots *R1*, *R2*, and *R3* because the sums of distances between robots and tasks are shortest by this method of allocation. As Figure 8(b) shows, the system will not reallocate tasks to robots until any one task is guided to its destination. When task *T2* reaches its destination, the system changes the latest task allocation strategy. Before the robot guide *T2* reach at its destination, robot *R1* execute the task *T5* and *R3* execute *T4*. After *T2* is guided to its destination, the robot changes to perform the task *T6*, while *R2* executes task *T5*. The snapshots of repeated auction method are presented in Figure 11.

Figure 10 Simulation results based on our approach (see online version for colours)



Notes: (a) $T = 0$: T1, T3, and T2 are respectively assigned to R1, R2, and R3. The destinations of tasks are, respectively, D1, D2, and D3. (b) $T = 62$: T1, T2, and T3 are assigned respectively to R1, R2, and R3. (c) $T = 110$: T3, T2, and T1 are assigned to R1, R2, and R3, respectively. (d) $T = 230$: T2, T3, and T1 are assigned respectively to R1, R2, and R3. (e) $T = 371$: T2, T3, and T1 are assigned respectively to R1, R2, and R3. R3 has reached around T1 and will guide T1 to the destination D1(20, –20). (f) $T = 475$: T3 and T2 are assigned to R1 and R2, respectively. R3 guides T1 to the destination D1(20, –20). (g) $T = 507$: R1 has reached around T3 and will guide T3 to the destination (–30, 100). T2 is assigned to R2. R3 guide T1 to the destination D1(20, –20). T4 walks freely. (h) $T = 667$: R1 guides T3 and to the destination (–30, 100). R2 has reached around T2 and will guide T2 to the destination (–60, 160). R3 guides T1 to the destination D1(20, –20). T4 walks freely. (i) $T = 705$: R1 guides T3 and to the destination (–30, 100). R2 guides T2 to the destination (–60, 160). R3 guides T1 to reach the destination D1(20, –20). T4 walks freely. (j) $T = 782$: R1 guides T3 to reach destination (–30, 100). R2 guides T2 to the destination (–60, 160). R3 assigns to T4. (k) $T = 969$: R1 assigns to T5. R2 guides T2 to destination (–60, 160). R3 has reached around T4 and will guide T4 to the destination (20, –180). T6 walks freely. (l) $T = 1{,}120$: R1 assigns to T5. R2 guides T2 to reach the destination (–60, 160). R3 has reached around T4 and will guide T4 to the destination (20, –180). T6 walks freely. (m) $T = 1{,}179$: R1 assigns to T6. R2 assigns to T5. R3 guides T4 to the destination (20, –180). (n) $T = 1{,}273$: R1 assigns to T6. R2 assigns to T5. R3 guides T4 to reach the destination (20, –180). (o) $T = 1{,}422$: R1 has reached around T6 and will guide T6 to the destination (180, –20). R2 assigns to T5. T4 has arrived at the destination (20, –180). (p) $T = 1{,}448$: R1 guides T6 to the destination (180, –20). R2 has reached around T5 and will guide T5 to the destination (180, –180). (q) $T = 1{,}789$: R1 guides T6 to the destination (180, –20). R2 guides T5 to reach the destination (180, –180). (r) $T = 1{,}836$: R1 guides T6 to reach the destination (180, –20).

Figure 10 Simulation results based on our approach (continued) (see online version for colours)



(m) (n) (o)

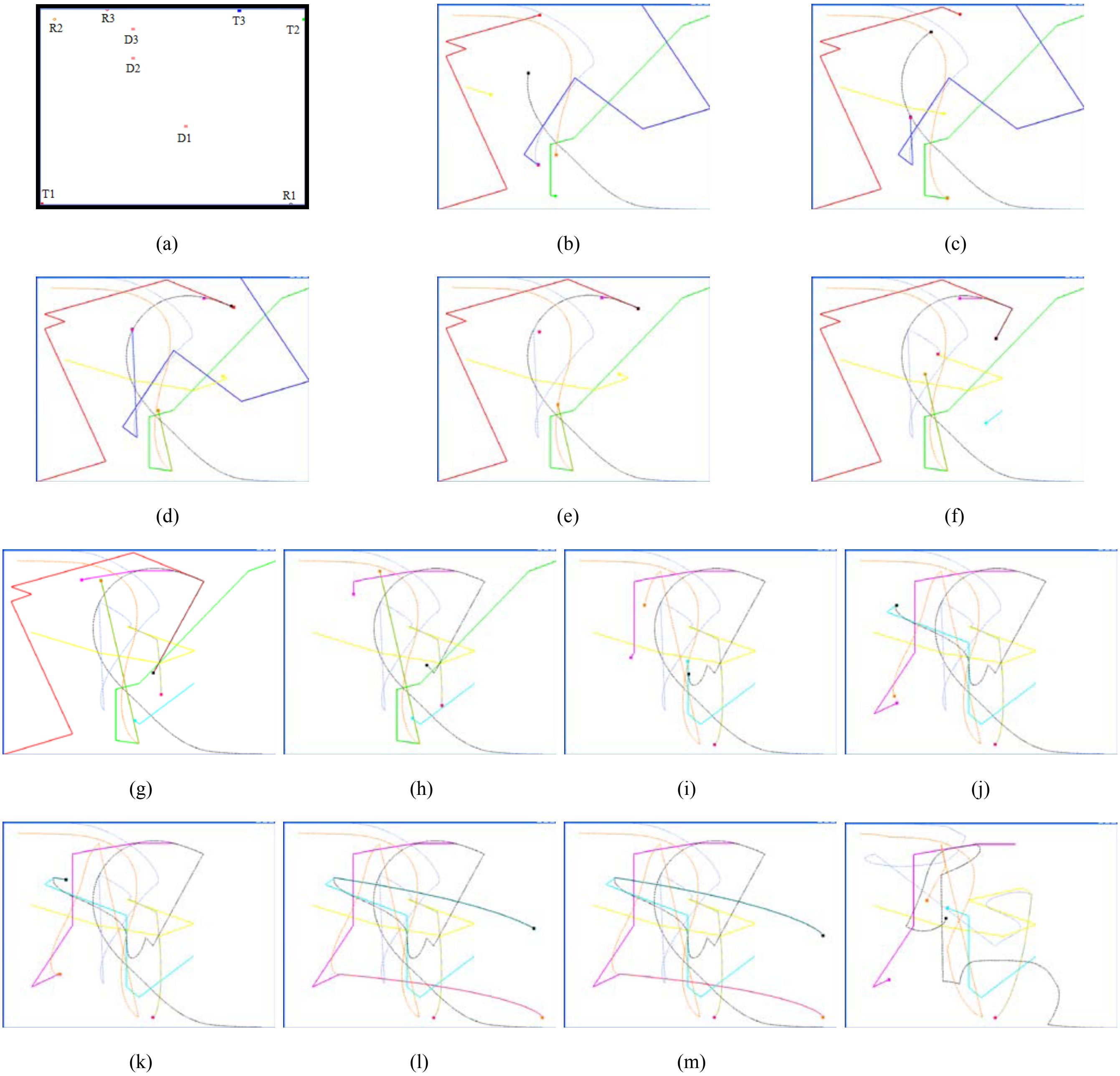
(p) (q) (r)

Notes: (a) $T = 0$: T1, T3, and T2 are respectively assigned to R1, R2, and R3. The destinations of tasks are, respectively, D1, D2, and D3. (b) $T = 62$: T1, T2, and T3 are assigned respectively to R1, R2, and R3. (c) $T = 110$: T3, T2, and T1 are assigned to R1, R2, and R3, respectively. (d) $T = 230$: T2, T3, and T1 are assigned respectively to R1, R2, and R3. (e) $T = 371$: T2, T3, and T1 are assigned respectively to R1, R2, and R3. R3 has reached around T1 and will guide T1 to the destination D1(20, –20). (f) $T = 475$: T3 and T2 are assigned to R1 and R2, respectively. R3 guides T1 to the destination D1(20, –20). (g) $T = 507$: R1 has reached around T3 and will guide T3 to the destination (–30, 100). T2 is assigned to R2. R3 guide T1 to the destination D1(20, –20). T4 walks freely. (h) $T = 667$: R1 guides T3 and to the destination (–30, 100). R2 has reached around T2 and will guide T2 to the destination (–60, 160). R3 guides T1 to the destination D1(20, –20). T4 walks freely. (i) $T = 705$: R1 guides T3 and to the destination (–30, 100). R2 guides T2 to the destination (–60, 160). R3 guides T1 to reach the destination D1(20, –20). T4 walks freely. (j) $T = 782$: R1 guides T3 to reach destination (–30, 100). R2 guides T2 to the destination (–60, 160). R3 assigns to T4. (k) $T = 969$: R1 assigns to T5. R2 guides T2 to destination (–60, 160). R3 has reached around T4 and will guide T4 to the destination (20, –180). T6 walks freely. (l) $T = 1{,}120$: R1 assigns to T5. R2 guides T2 to reach the destination (–60, 160). R3 has reached around T4 and will guide T4 to the destination (20, –180). T6 walks freely. (m) $T = 1{,}179$: R1 assigns to T6. R2 assigns to T5. R3 guides T4 to the destination (20, –180). (n) $T = 1{,}273$: R1 assigns to T6. R2 assigns to T5. R3 guides T4 to reach the destination (20, –180). (o) $T = 1{,}422$: R1 has reached around T6 and will guide T6 to the destination (180, –20). R2 assigns to T5. T4 has arrived at the destination (20, –180). (p) $T = 1{,}448$: R1 guides T6 to the destination (180, –20). R2 has reached around T5 and will guide T5 to the destination (180, –180). (q) $T = 1{,}789$: R1 guides T6 to the destination (180, –20). R2 guides T5 to reach the destination (180, –180). (r) $T = 1{,}836$: R1 guides T6 to reach the destination (180, –20).

Similarly, as Figure 8(c) shown, at first time step, according to the combinatorial cost table, task *T1*, *T2* and *T3* are assigned to *R1*, *R2* and *R3*, respectively based on global optimal method. After 10 time steps, at $T = 11$, the whole system reassigns robot *R1*, *R2* and *R3* to task *T1*, *T3* and *T2* respectively, because the entire distance of such combinatorial strategy is shortest than others. $T = 90$, 231, 570, 850, 882 and 1,020 are the same as $T = 11$. The snapshots of repeated auction method are presented in Figure 12.

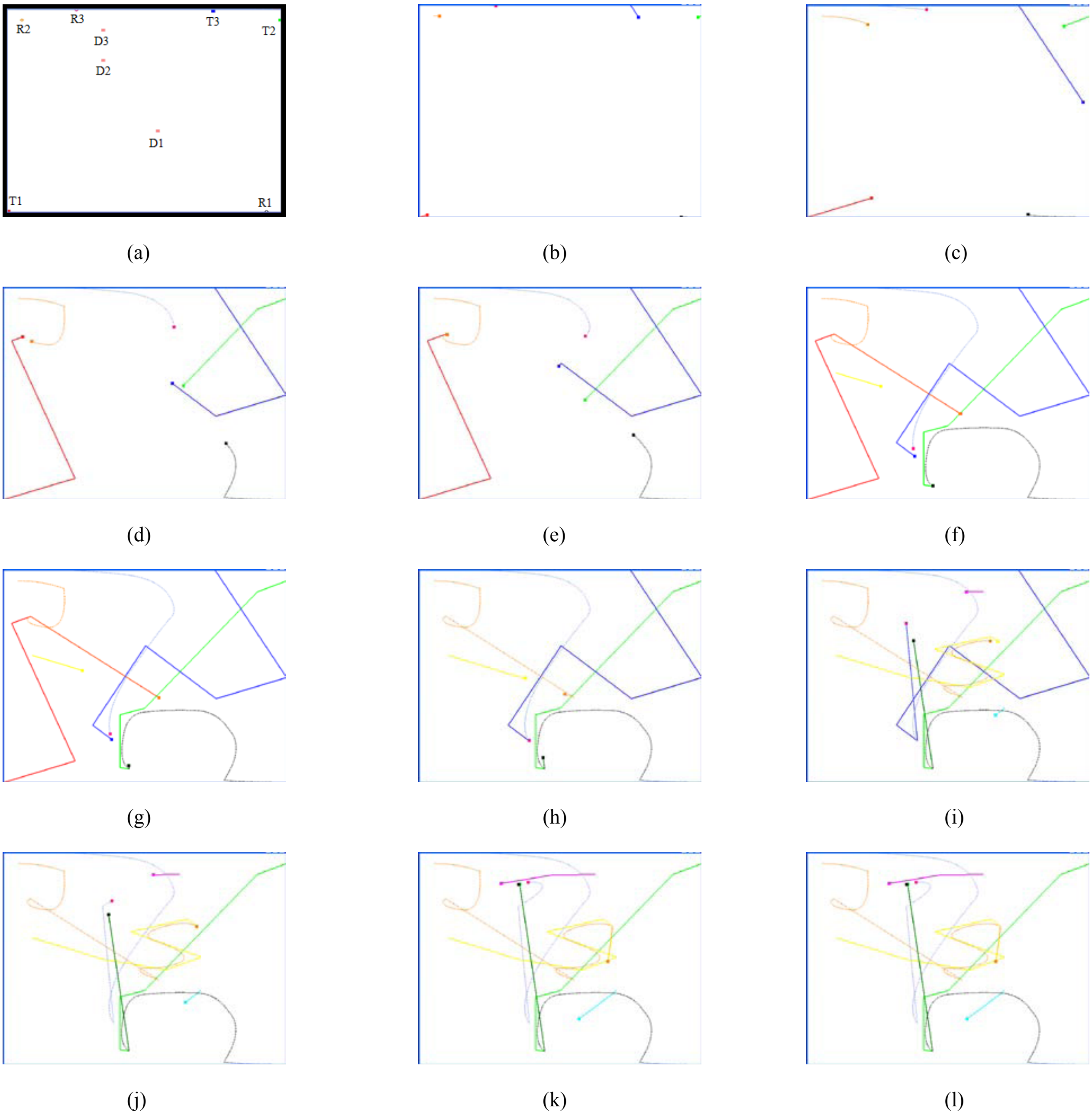
Results show the condition under which a robot assigns a task during simulation. As the figures show, it is apparent that robots often change tasks to perform according to distance, but not as frequently as we expected. Figure 13 shows the time steps that robots reach around tasks and guide tasks to destinations. Figure 14 shows the total consumed time steps that robots reach around the first three tasks and all tasks, and guide the first three tasks and all tasks to the destinations. Simulation results show that the total number of time steps for robots reach around tasks is *3,134* using our method. The first three tasks need only *1,545* time steps. The total time steps by which robots guide the first three tasks and all tasks to the destination are *2,607* and *7,505*. Similarly, for the repeated auction method, the total number of time steps for robots reach around tasks is *3,692*. The increased time steps are 458 more than those obtained using our approach. For the first three tasks, it needs 2,018 time steps. The total time steps by which robots guide the first three tasks and all tasks to the destination are 3,007 and 7897. For global optimal assignment method, the total number of time steps for robots reach around tasks is 3,293, while for the first three tasks it needs 1,840 time steps. The total time steps by which robots guide the first three tasks and all tasks to the destination are 2,472 and 7,165. The detailed improved performance of our approach relative to the existing task allocation algorithms is shown in Figure 14.

Figure 11 Simulation results based on repeated auction method, (see online version for colours)



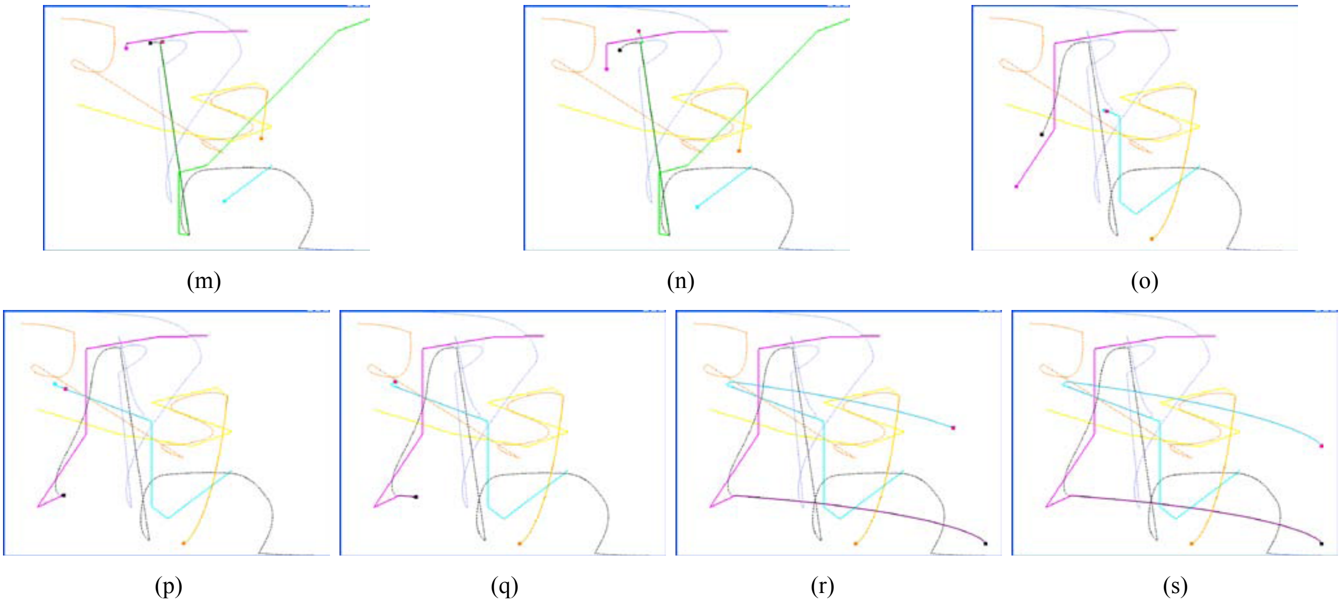
Notes: (a) $T = 0$: T1, T2, and T3 are respectively allocated to robot R1, R2, and R3. (b) $T = 534$: T1 and T2 are respectively allocated to robot R1 and R2. R3 has reached around T3 and will guide T3 to the destination (–60, 100). T4 move freely. (c) $T = 657$: T1 is allocated to robot R1. R2 has reached around T2 and will guide T2 to the destination (–60, 160). R3 guides T3 to the destination (–60, 100). T4 move freely. (d) $T = 814$: T1 is allocated to robot R1. R2 guides T2 to the destination (–60, 160). R3 guides T3 to reach the destination (–60, 100). T4 and V5 move freely. (e) $T = 827$: R1 has reached around T1 and will guide T1 to the destination (20, –40). R2 guides T2 to the destination (–60, 160). T4 are assigned to R3, T5 walks freely. (f) $T = 909$: R1 guides T1 to the destination (20, –40). R2 guides T2 to the destination (–60, 160). R3 has reached around T4 and will guide T4 to the destination (20, –180). T5 and T6 walk freely. (g) $T = 1084$: R1 guides T1 to reach the destination (20, –40). R2 guides T2 to the destination (–60, 160). R3 guides T4 to the destination (20, –180). T5 and T6 walk freely. (h) $T = 1109$: T5 is assigned to R1. R2 guide T2 to reach the destination (–60, 160). R3 guides T4 to the destination (20, –180). T6 walks freely. (k) $T = 1{,}203$: T6 is assigned to R1. T5 is assigned to R2. R3 guide T4 to reach the destination (20, –180). (j) $T = 1{,}447$: R1 has reached around T6 and will guide T6 to the destination (180, –20). T5 is assigned to R2. (k) $T = 1{,}468$: R1 guides T6 to the destination (180, –20). R2 has reached around T5 and will guide V5 to the destination (180, –180). (l) $T = 1830$: R1 guides T6 to the destination (180, –20). R2 guides T5 to reach the destination (180, –180). (m) $T = 1{,}857$: R1 guide T6 to reach the destination (180, –20).

Figure 12 Simulation results based on global optimal method (see online version for colours)



Notes: (a) $T = 0$: T1, T2, and T3 are respectively assigned to R1, R2, and R3. The destinations of tasks are, respectively, D1, D2, and D3. (b) $T = 11$: T1, T3, and T2 are assigned respectively to R1, R2, and R3. (c) $T = 90$: T3, T1, and T2 are assigned to R1, R2, and R3, respectively. (d) $T = 231$: T2, T1, and T3 are assigned respectively to R1, R2, and R3. (e) $T = 256$: T2, T1, and T3 are assigned respectively to R1, R2, and R3. R2 has reached around T1 and will guide T1 to the destination D1(20, –40). (f) $T = 562$: T1 and T3 are assigned to R2 and R3, respectively. R1 guides T2 to the destination D1(20, –40), R1 has captured T2 and will guide T2 to the destination (–60, 160). (g) $T = 570$: R1 guides T2 to the destination (–60, 160), T1 has arrived at (20, –40) and R2 will assign to T4, R3 assigns to T3. (h) $T = 589$: R1 guides T2 to (–60, 160), R2 assigns to T4, R3 has reached around T3 and will guide T3 to (–30, 100). (i) $T = 882$: R1 guides T2 to (–60, 160), R2 assigns to T4, T3 has arrived at (–60, 100) and R3 will assign to T5. (j) $T = 903$: R1 guides T2 to (–60, 160), R2 has reached around T4 and will guide T4 to (20, –180), R3 assigns to T5. (k) $T = 981$: R1 assigns to T5, R2 guides T4 to (20, –180), R3 assigns to T2. (l) $T = 995$: R1 assigns to T5, R2 guides T4 to (20, –180), R3 has reached around T2 and will guide T2 to (–60, 160). (m) $T = 1{,}020$: R1 assigns to T5. R2 guides T4 to (20, –180). R3 guides T2 to (–600, 160). (n) $T = 1{,}208$: R1 assigns to T5. T4 has arrived at (20, –180), R3 assigns to T6. (o) $T = 1{,}339$: R1 has captured T5 and will guide T5 to (180, –180), R3 assigns to T6. (p) $T = 1{,}361$: R1 guides T5 to (180, –180), R3 has reached around T6 and will guide T6 to (180, –20). (q) $T = 1{,}709$: T5 has arrived at (180, –180), R3 guides T6 to (180, –20). (s) $T = 1{,}776$: R3 guides T6 to (180, –20).

Figure 12 Simulation results based on global optimal method (continued) (see online version for colours)



Notes: (a) $T = 0$: T1, T2, and T3 are respectively assigned to R1, R2, and R3. The destinations of tasks are, respectively, D1, D2, and D3. (b) $T = 11$: T1, T3, and T2 are assigned respectively to R1, R2, and R3. (c) $T = 90$: T3, T1, and T2 are assigned to R1, R2, and R3, respectively. (d) $T = 231$: T2, T1, and T3 are assigned respectively to R1, R2, and R3. (e) $T = 256$: T2, T1, and T3 are assigned respectively to R1, R2, and R3. R2 has reached around T1 and will guide T1 to the destination D1(20, –40). (f) $T = 562$: T1 and T3 are assigned to R2 and R3, respectively. R1 guides T2 to the destination D1(20, –40), R1 has captured T2 and will guide T2 to the destination (–60, 160). (g) $T = 570$: R1 guides T2 to the destination (–60, 160), T1 has arrived at (20, –40) and R2 will assign to T4, R3 assigns to T3. (h) $T = 589$: R1 guides T2 to (–60, 160), R2 assigns to T4, R3 has reached around T3 and will guide T3 to (–30, 100). (i) $T = 882$: R1 guides T2 to (–60, 160), R2 assigns to T4, T3 has arrived at (–60, 100) and R3 will assign to T5. (j) $T = 903$: R1 guides T2 to (–60, 160), R2 has reached around T4 and will guide T4 to (20, –180), R3 assigns to T5. (k) $T = 981$: R1 assigns to T5, R2 guides T4 to (20, –180), R3 assigns to T2. (l) $T = 995$: R1 assigns to T5, R2 guides T4 to (20, –180), R3 has reached around T2 and will guide T2 to (–60, 160). (m) $T = 1,020$: R1 assigns to T5. R2 guides T4 to (20, –180). R3 guides T2 to (–600, 160). (n) $T = 1,208$: R1 assigns to T5. T4 has arrived at (20, –180), R3 assigns to T6. (o) $T = 1,339$: R1 has captured T5 and will guide T5 to (180, –180), R3 assigns to T6. (p) $T = 1,361$: R1 guides T5 to (180, –180), R3 has reached around T6 and will guide T6 to (180, –20). (q) $T = 1,709$: T5 has arrived at (180, –180), R3 guides T6 to (180, –20). (s) $T = 1,776$: R3 guides T6 to (180, –20).

Therefore, it is demonstrated that the consumed time steps which under our proposed method for robots reach around and guide the first three tasks to their destinations are fewer than under repeated auction methods. While for accomplishing all tasks, the consumed time steps by utilising global optimal assignment method are less than our approach, because of our proposed approach is an approximation global optimal allocation method, which is a suboptimal allocation approach. That is for the whole distributed multi-robot coordination, the efficiency to accomplish the given tasks is improved greatly using our proposed dynamical-sequential task allocation and reallocation method.

5.5 *Communication costs and computation times*

An important strength of our proposed task allocation and reallocation method is the ability to address changing conditions efficiently. Our method does not rely on the initial allocation solution. It can perform task reallocation according to variable solutions. Therefore, the distributed multi-robot coordination system is highly robust for environment changing, including robot malfunction and communication failure. Consequently, the method presented in this paper enables robots to address the dynamical environment in an opportunistic and adaptive manner. Communication costs and computational times using the two methods described above are presented in Table 1 and Table 2, where M represents the number of robots, N signifies the number of robots, T denotes the number of time steps, and T_0 stands for the time unit used for calculating the distance from one robot to one task.

Table 1 Communication costs

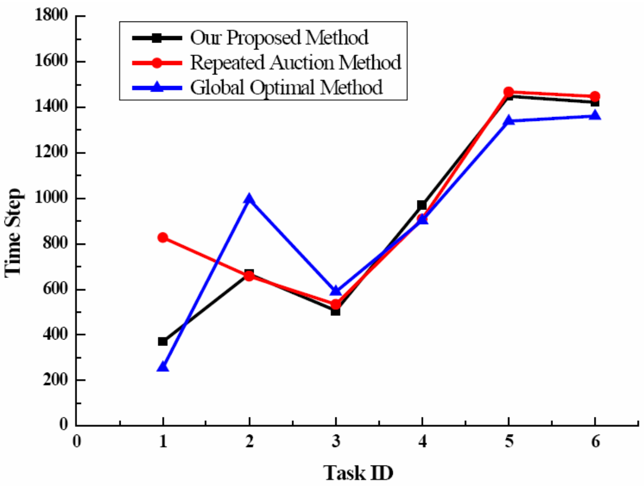
Dynamical sequential	*Repeated auctions*	*Global optimal*
Variable	$2 * M * (M - 1) * (N - M + 1)$	$2 * N * (N - 1) * T$

Table 2 Computation times

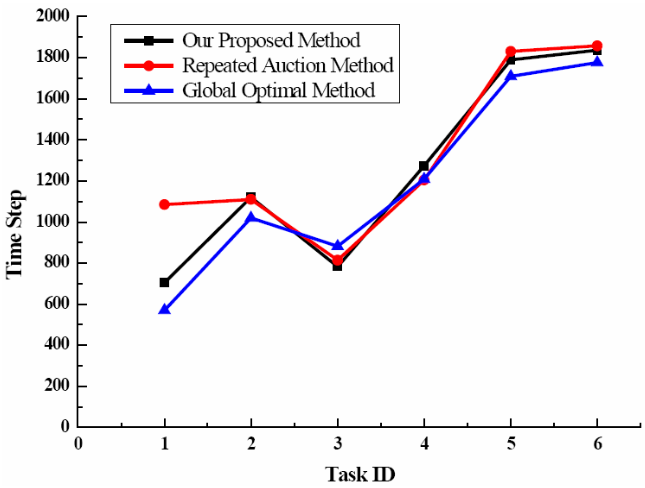
Dynamical sequential	*Repeated auctions*	*Global optimal*
Variable	$M * M * T_0 + (N - M) * T_0$	$N * N * T * T_0$

Communication costs and computational times of dynamical-sequential task allocation method vary according to time because of implementation of body expansion behaviour for robots to select tasks. That is, if the distance between robot and task is less than $D2_{Threshold}$, then the robot only plans a path to the latest assigned task. Consequently, it is more conducive to reduce the numerous computational times to plan paths and calculate distances for the entire system. Because communication is used for transmission of task selection information between robots, communication costs are greatly decreased compared to the existing investigated methods at each time step, especially compares with global optimal method. Figure 15 and Figure 16 show the communication costs and computational times for the simulation example above.

Figure 13 Executed time step, (a) robot approaches tasks* (b) tasks at destinations** (see online version for colours)



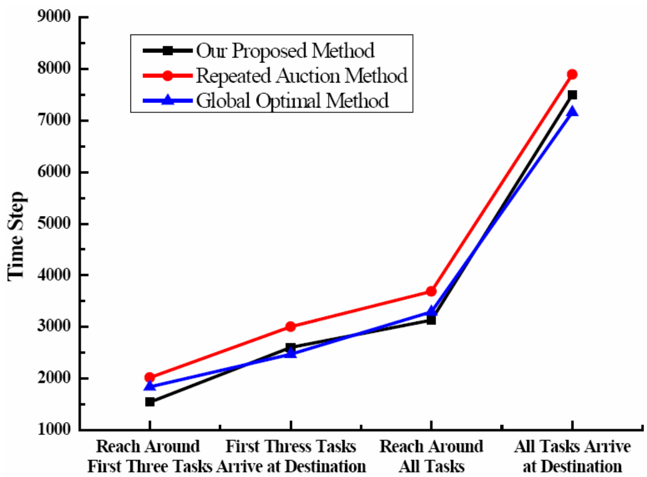
(a)



(b)

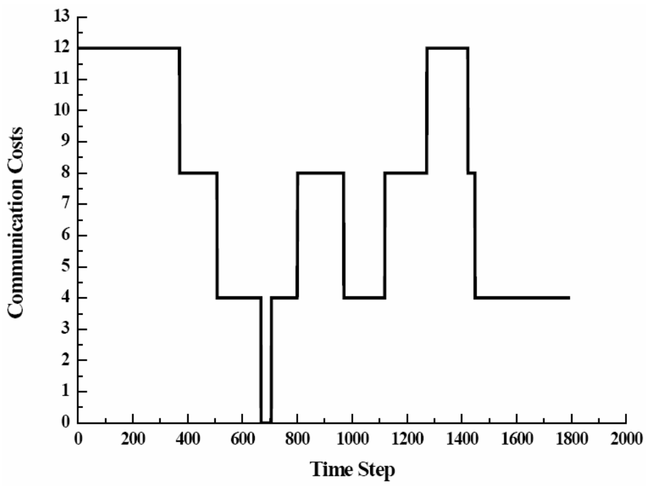
Notes: *Based on our proposed method, robots reach around task *T1*, *T2*, *T3*, *T4*, *T5* and *T6* at time step T = 371, 667, 507, 969, 1,448 and 1,422, respectively. Similarly, for repeated auction method, robots reach around task *T1*, *T2*, *T3*, *T4*, *T5* and *T6* at time step T = 827, 657, 534, 909, 1,468 and 1,447, respectively. For global optimal method, robots reach around task *T1*, *T2*, *T3*, *T4*, *T5* and *T6* at time step T = 256, 995, 589, 903, 1,339 and 1,361, respectively.
**Based on our proposed method, robots guide task *T1*, *T2*, *T3*, *T4*, *T5* and *T6* to its destination at time step T = 705, 1,120, 782, 1,273, 1,789 and 1,836, respectively. Similarly, for repeated auction method, robots guide task *T1*, *T2*, *T3*, *T4*, *T5* and *T6* to its destination at time step T = 1,084, 1,109, 814, 1,203, 1,830 and 1,857, respectively. For global optimal method, robots guide task *T1*, *T2*, *T3*, *T4*, *T5* and *T6* to its destination at time step T = 570, 1,020, 882, 1,208, 1,709 and 1,776, respectively.

Figure 14 Total consumed time steps (see online version for colours)

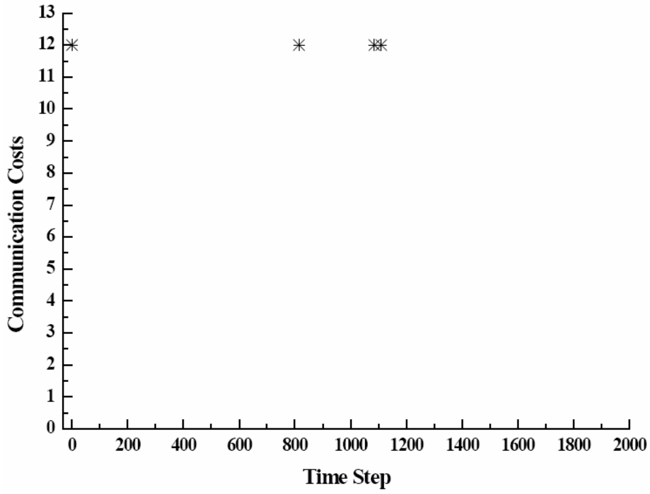


Notes: Based on our proposed method, robots consumed 1,545, 2,607, 3,134 and 7,505 time steps to reach around the first three tasks, guide the first three tasks to its destination, reach around all tasks and guide all tasks to its destination, respectively. Similarly, for repeated auction method, robots consumed 2,018, 3,007, 3,692 and 7,897 time steps to reach around the first three tasks, guide the first three tasks to its destination, reach around all tasks and guide all tasks to its destination, respectively. For global optimal method, robots consumed 1,840, 2,472, 3,293 and 7,165 time steps to reach around the first three tasks, guide the first three tasks to its destination, reach around all tasks and guide all tasks to its destination, respectively.

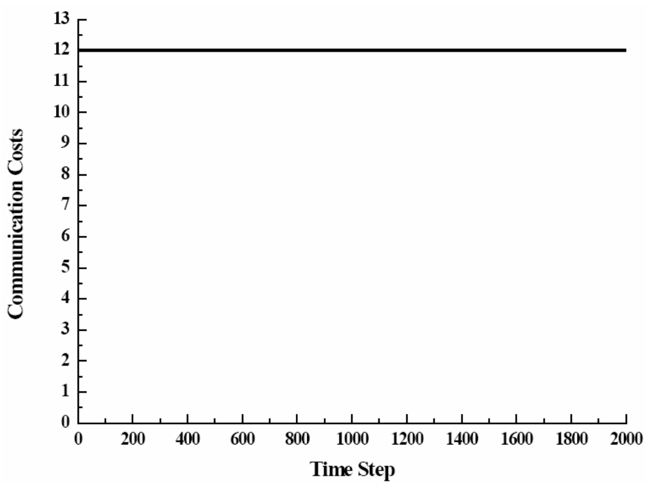
Figure 15 Communication costs, (a) proposed method* (b) repeated auction method** (c) global optimal method***



(a)



(b)



(c)

Notes: *Using the proposed method, only when all distances between robots and tasks are greater than $D2_{Threshold}$ does the system require 12 unit communication costs. Otherwise, for most of the simulation time, the consumed communication costs are much less than 12 units; at some time steps, the consumed time steps are 0.

** Based on the repeated auction method, every time robots reassign tasks require consumption of 12 unit communication costs, such as at time step $T = 0$, 1,084, 1,109 and 1,203.

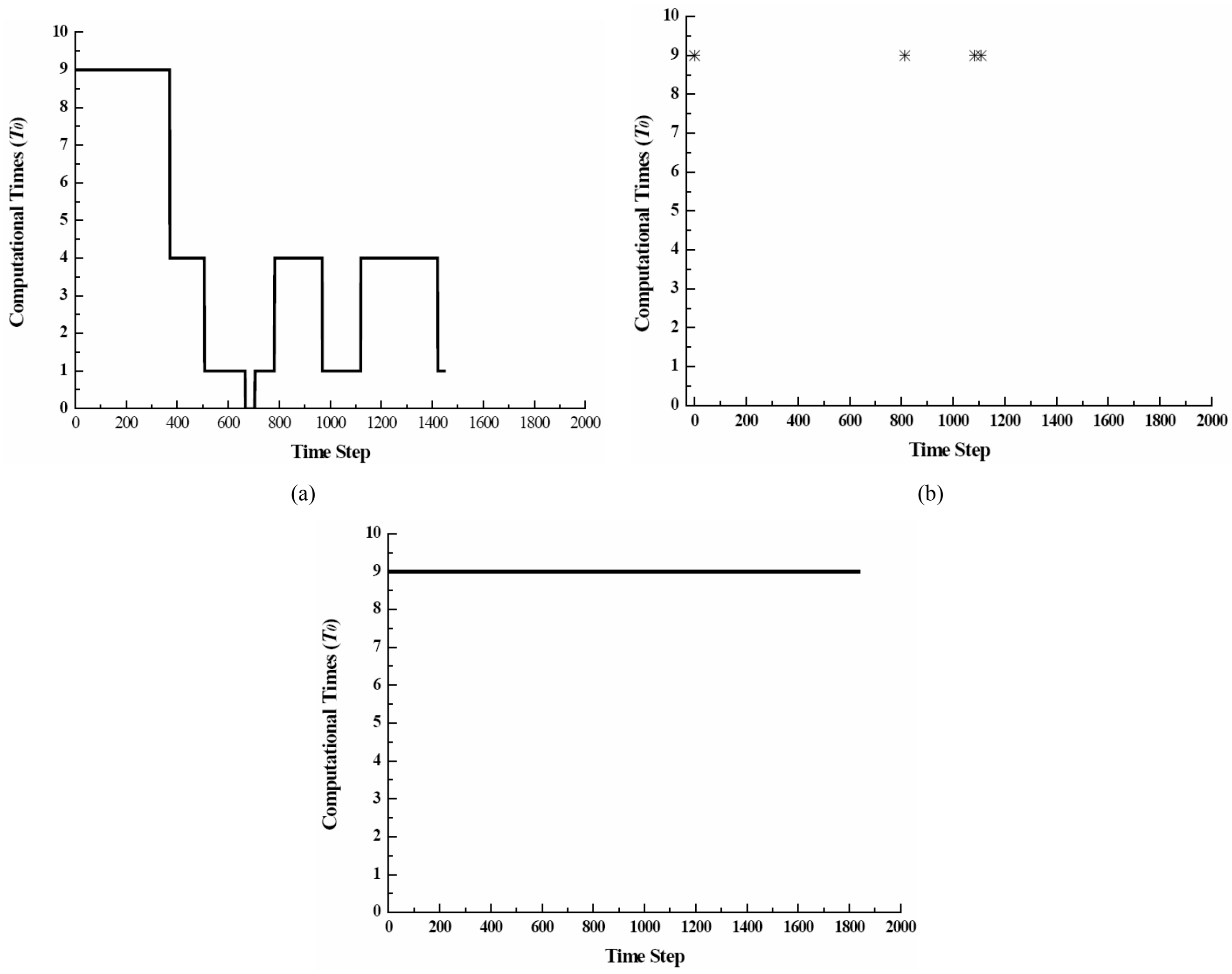
***Based on global optimal method, every time step robots require consumption of 12 unit communication costs.

5.6 Discussion

As discussed above, the implemented $D1_{Threshold}$ and $D2_{Threshold}$ are assigned as a priori values. Hereinafter, we specifically examine the study of the influence of threshold setting on our proposed dynamical-sequential task allocation and reallocation method. Actually, we have simulated large and small distance threshold values of various kinds and their combinations (Li et al., 2011c).

Figures 17(a) to 21(a) present results of completed time steps, the consumed time steps, computational times, communication costs, and the numbers of changing assigned times under large and small distance threshold values of various kinds and their combinations. As the figures show, it is apparent that values of setting large distance and small distance thresholds exert a great influence on performances of dynamical-sequential task allocation and reallocation methods. Especially, those performance parameters including completed time steps, the total consumed time steps, computational times and communication costs fluctuate markedly when the values of large and small distance thresholds are large.

Figure 16 Computation times, (a) proposed method* (b) repeated auction method** (c) global optimal method***

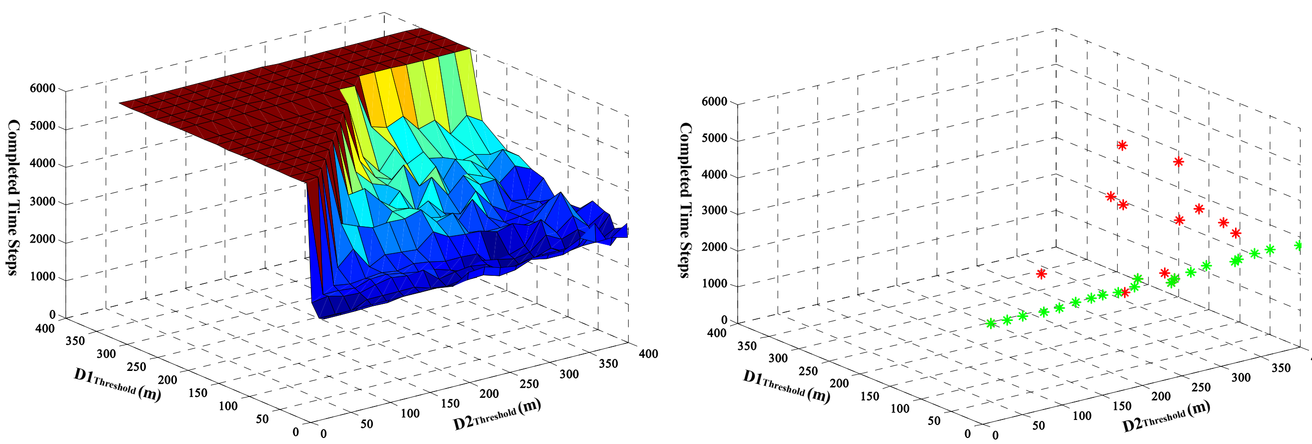


(a) (b)

(c)

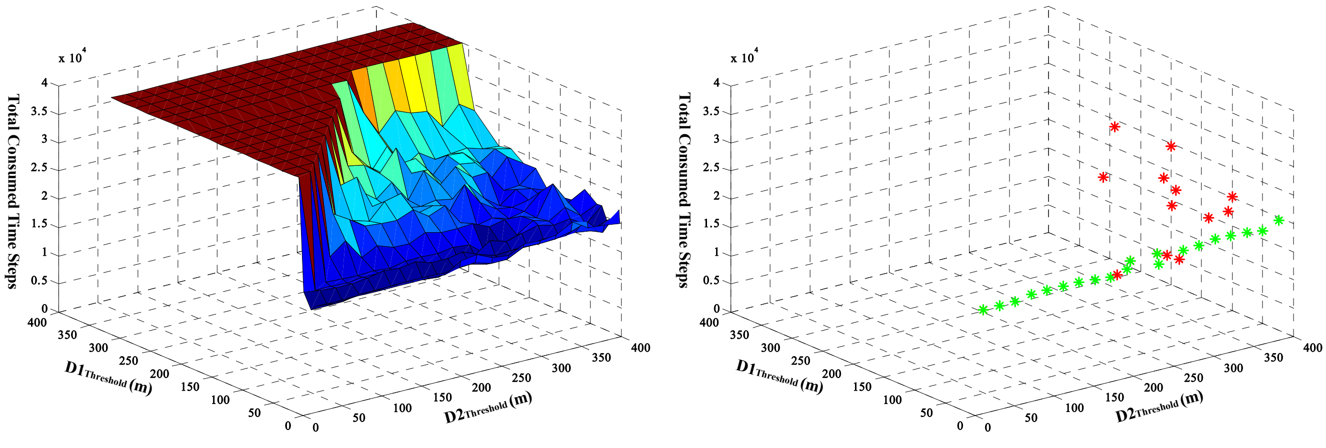
Notes: *As with communication costs, using the proposed method, only when all distances between robots and tasks are greater than $D2_{Threshold}$ did the system require 9 T_0 computational times. Otherwise, most simulation times, the consumed communication costs were much less than 9 T_0; even at some time steps, the consumed time step is 0.
**Using the repeated auction method every time robots reassign tasks consumed 9 T_0 to compute distances and plan paths.
***Using the global optimal method every time step robots consumed 9 T_0 to compute distances and plan paths.

Figure 17 Completed time steps (see online version for colours)



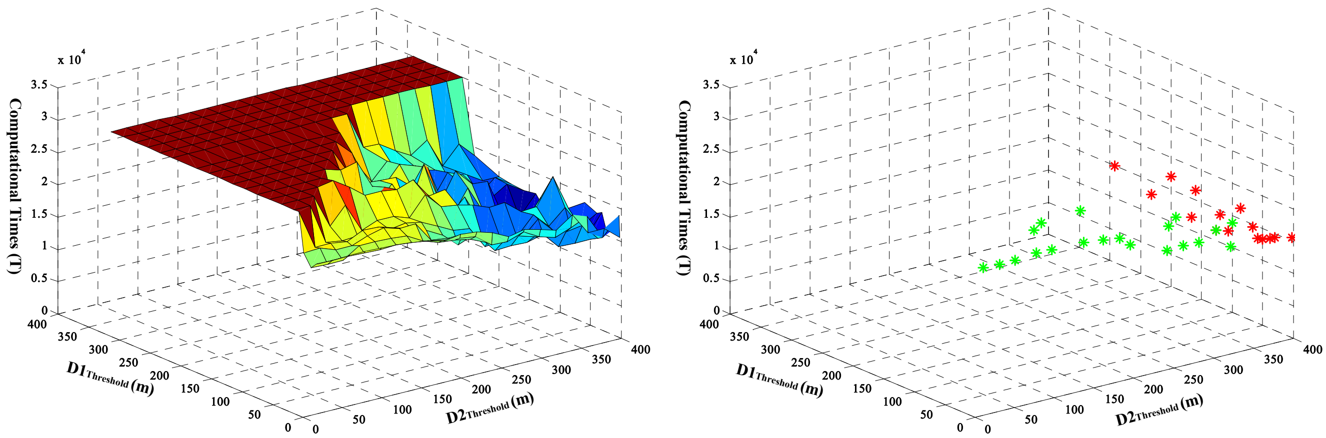
(a) (b)

Figure 18 Total consumed time steps (see online version for colours)



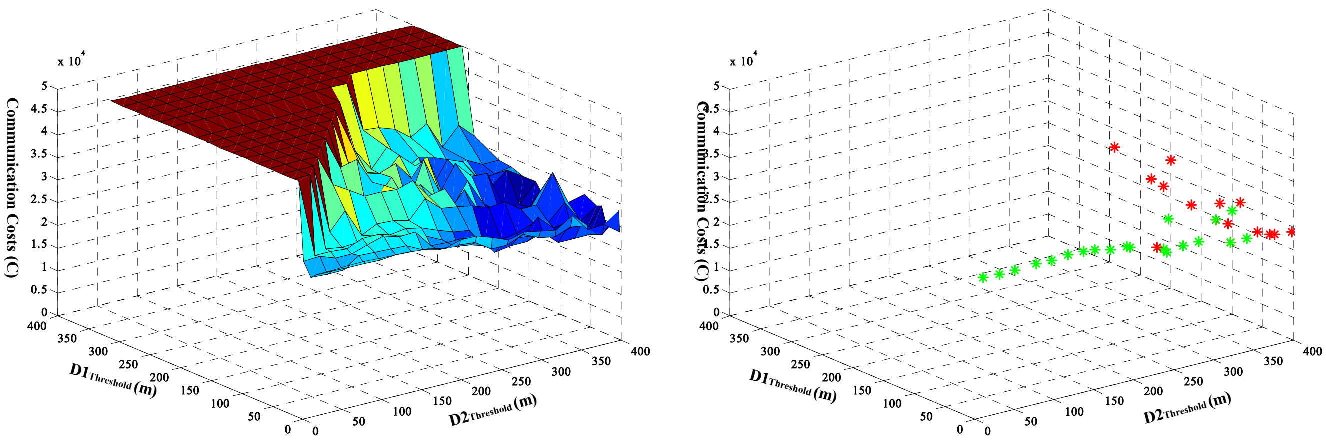
(a) (b)

Figure 19 Computational times (see online version for colours)



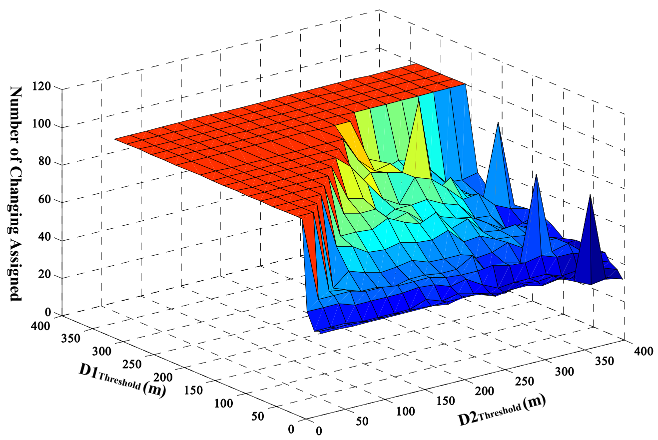
(a) (b)

Figure 20 Communication costs (see online version for colours)

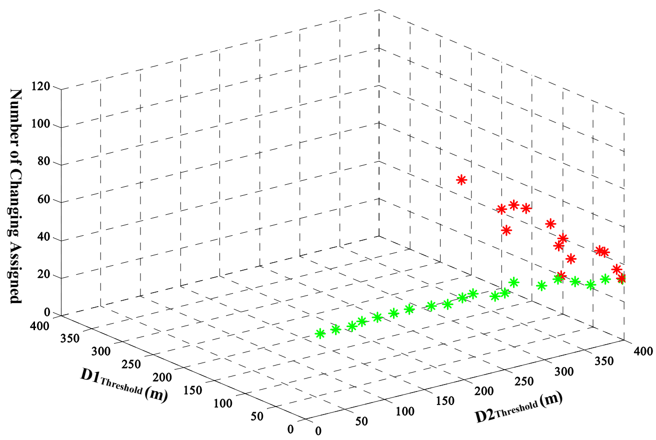


(a) (b)

Figure 21 Number of changing assigned (see online version for colours)



(a)



(b)

Simulation results suggest that it is better to allocate large and small distance thresholds to a small value according to the environment area, the large distance threshold is about one-tenth of environment's length and width, the small distance threshold is less than half of the large distance threshold. It is particularly interesting that if the $D2_{Threshold}$ is settled as constant, then the $D1_{Threshold}$ can vary. The optimal values of completed time steps, total consumed time steps, computational times, communication costs and number of changing assigned times appear when small distance threshold is small, as Figures 17(b) to 21(b) show, as presented by green points. Also, if the $D1_{Threshold}$ is set as constant, then $D2_{Threshold}$ changes. The optimal values of performance appear when the large distance threshold is great, as presented in Figures 15(b) to 19(b) by red points. We believe that the results are important for us to choose and to adjust the values of threshold to adapt to different environment scales.

6 Conclusions

A distributed multi-robot coordination system based on the method of dynamical-sequential task allocation and reallocation is presented in this paper. We propose such a method to reallocate tasks to robots for a dynamical moving task assignment problem. Our proposed method for multiple robots coordination system is based on multi-round negotiation and body expansion behaviour. To demonstrate the validity and efficiency of the proposed approach, various experiments are conducted using computer simulations. We simulate a kind of general approach market-based repeated auction method in the same situation compared with the proposed approach. Simulation results show that the proposed task allocation and reallocation method has greater validity and efficiency than the general task allocation method. Moreover, it is more conducive to reduction of the numerous computational times and communication costs compared to existing investigated task assignment methods.

However, the disadvantages of our algorithm are that it still takes a long time for negotiation and communication, and requires certain computational times. Moreover, the dynamical-sequential method is an approximate global optimal allocation method that is a suboptimal allocation approach. Therefore, it is extremely well suited for small and medium scale distributed multi-robot coordination systems. To resolve such problems, it is better to combine our dynamical-sequential method and global optimal method for large-scale distributed multi-robot coordination, especially to reduce the negotiation time and communication costs. In future works, we will improve the proposed algorithms for a large-scale multi-robot coordination system and implement our approaches to a real robot system, such as employing our methods for guidance service system using multiple robots to guide customers in shopping malls, museums, and exhibitions.

Acknowledgements

The present research was supported (in part) through the Global COE Programme, 'Global Center of Excellence for Mechanical System Innovation' by the Ministry of Education, Culture, Sports, Science and Technology, Japan. And this paper is a revised and expanded version of our international conference paper entitled 'Dynamical task allocation and reallocation based on body expansion behaviour for multi-robot coordination system' which presented at the 2011 IEEE International Conference on Mechatronics and Automation, August 7–10, 2011, Beijing, China.

References

Asama, H., Ozaki, K., Matsumoto, A., Ishida, Y. and Endo, I. (1992) 'Development of task assignment system using communication for multiple autonomous robots', *Journal of Robotics and Mechatronics*, Vol. 4, No. 2, pp.122–127.

Dias, M.B., Zlot, R., Kalra, N. and Stentz, A. (2006) 'Market-based multirobot coordination: a survey and analysis', *Proceedings of the IEEE*, Vol. 94, No. 7, pp.1257–1270.

Fujiki, T., Kawabata, K. and Asama, H. (2007) 'Adaptive action selection of body expansion behaviour in multi-robot system using communication', *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 11, No. 2, pp.142–147.

Gerkey, B.P. and Mataric, M.J. (2002) 'Auction methods for multi-robot coordination', *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, pp.758–768.

Li, G., Tamura, Y. and Asama, H. (2011a) 'Dynamical task allocation and reallocation based on body expansion behavior for multi-robot coordination system', *Proceedings of 2011 IEEE International Conference on Mechatronics and Automation (ICMA2011)*, pp.537–542.

Li, G., Tamura, Y. and Asama, H. (2011b) 'A comparative study of dynamical sequential and global optimal task reallocation methodology for distributed multi-robot system', *Proceedings of the 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI2011)*, pp.307–312.

Li, G., Tamura, Y. and Asama, H. (2011c) 'Influence analysis of setting thresholds on dynamical sequential task allocation and reallocation methodology', *Proceedings of the 2011 IEEE/SICE International Symposium on System Integration (SII2011)*, pp.1165–1170.

Nanjanath, M. and Gini, M. (2010) 'Repeated auction for robust task execution by a robot team', *Robotics and Autonomous System*, Vol. 58, No. 7, pp.900–909.

Ozaki, K., Asama, H. and Endo, I. (1997) 'Distributed and cooperative object pushing by multiple mobile robots based on communication', *Advanced Robotics*, Vol. 11, No. 5, pp.501–517.

Paola, D.D., Gasparri, A., Giovanni, D.N. and Lewis, F.L. (2011) 'Decentralized task sequencing and multiple mission control for heterogeneous robotic networks', *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp.4467–4473.

Parker, L.E. (1992) 'Adaptive action selection for cooperative agent teams', *Proceedings of the Second International Conference of Simulation of Adaptive Behavior*, pp.442–450.

Parker, L.E. (1994a) 'ALLIANCE: an architecture for fault tolerance, cooperative control of heterogeneous mobile robots', *Proceedings of 1994 IEEE/RSJ/GI, International Conference on Intelligent Robotics Systems (IROS 94)*, pp.776–783.

Parker, L.E. (1994b) 'Fault tolerance multi-robot cooperation', MIT Artificial Intelligence Lab Videotape AIV-9.

Parker, L.E. (1994c) 'Heterogeneous multi-robot cooperation', PhD dissertation, Massachusetts Institute of Technology, Artificial Intelligence Lab.

Parker, L.E. (1997) 'L-ALLIANCE: task-oriented multi-robot learning in behaviour-based system', *Advance Robotics*, Vol. 11, No. 4, pp.305–322.

Parker, L.E. (1998) 'ALLIANCE: an architecture for fault tolerance multi-robot cooperation', *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 2, pp.220–240.

Parker, L.E. and Tang, F. (2006) 'Building multi-robot coalitions through automated task solution synthesis', *Proceedings of IEEE*, Vol. 94, No. 7, pp.1289–1305.

Sariel, S. and Balch, T. (2005a) 'Robust multi-robot coordination in noisy and dangerous environment', GVU Technical Report: GIT-GVU-05-17, Georgia Institute of Technology.

Sariel, S. and Balch, T. (2005b) 'Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments', *Proceedings of AAAI Workshop: Integrating Planning into Scheduling*, pp.27–33.

Sariel, S. and Balch, T. (2008) 'A distributed multi-robot cooperation framework for real time task achievement', *Proceedings of Distributed Autonomous Robotic Systems*, Vol. 8, pp.187–196.

Sariel, S., Balch, T. and Stack, J. (2008) 'Empirical evaluation of auction-based coordination of AUVs in a realistic simulated mine countermeasure task', *Proceedings of Distributed Autonomous Robotic Systems*, Vol. 8, pp.197–206.

Stentz, A. and Dias, M.B. (2006) 'A free market architecture for coordinating multiple robots', *Robotics Inst.*, Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep.

Sujit, P.B. and Beard, R. (2007) 'Distributed sequential auction for multiple UAV task allocation', *Proceedings of the 2007 American Control Conference*, pp.3955–3960.

Sujit, P.B., George, J.M. and Beard, R. (2008) 'Multiple UAV task allocation using particle swarm optimization', *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA2008-6837.

Tang, F. and Parker, L.E. (2007) 'A complete methodology for generating multi-robot task solutions using ASyMTRe-D and market-based task allocation', *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp.3351–3358.

Turra, D., Pollini, L. and Innocenti, M. (2004a) 'Fast unmanned vehicles task allocation with moving targets', *Proceedings of the 43rd IEEE Conference on Decision and Control*, pp.4280–4285.

Turra, D., Pollini, L. and Innocenti, M. (2004b) 'Real-time unmanned vehicle task allocation with moving targets', *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA2004-5253.

Wawerla, J. and Vaughan, R.T. (2010) 'A fast and frugal method for team-task allocation in a multi-robot transportation system', *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp.1432–1437.

Werger, B.B. and Mataric, M.J. (2000) 'Broadcast of local eligibility for multi-target observation', *Proceedings of Distributed Autonomous Robotic Systems*, Vol. 4, pp.347–356.

Zheng, X., Koenig, M.S. and Tovey, C. (2006) 'Improving sequential single-item auction', *Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems*, pp.231–237.